

Bonus
CD-ROM
inside!



 sitepoint

Firefox Secrets



By Cheah Chu Yeow

A Need-To-Know Guide

Firefox Secrets (Excerpt)

Thank you for downloading this excerpt from Cheah Chu Yeow's book, *Firefox Secrets*, published by SitePoint.

This excerpt includes the Summary of Contents, Information about the Author, Editors and SitePoint, Table of Contents, the Introduction, Chapter 2, and selected portions of Chapter 6 and Chapter 7.

We hope you find this information useful in evaluating this book.

[For more information or to order, visit sitepoint.com](http://sitepoint.com)

Summary of Contents of this Excerpt

Introduction	ix
2. Essential Browsing Features.....	25
6. Tips, Tricks, and Hacks (Excerpt)	173
7. Web Development Nirvana (Excerpt)	209
Index.....	283

Summary of Additional Book Contents

1. Introducing Your New Favorite Web Browser.....	1
3. Revisiting Web Pages.....	71
4. Dressing Up Firefox	105
5. Personalizing Firefox.....	115
6. Tips, Tricks and Hacks	173
7. Web Development Nirvana.....	209
8. Living On The Edge	263

Firefox Secrets

by Cheah Chu Yeow

Firefox Secrets

by Cheah Chu Yeow

Copyright © 2005 SitePoint Pty. Ltd.

Managing Editor: Simon Mackie

Technical Editor: Nigel McFarlane

Expert Reviewer: Asa Dotzler

Editor: Georgina Laidlaw

Printing History:

First Edition: June 2005

Index Editor: Bill Johncocks

Cover Designer: Julian Carroll

Cover Illustrator: Lucas Licata

Notice of Rights

All rights reserved. No part of this book may be reproduced, stored in a retrieval system or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical articles or reviews.

Notice of Liability

The author and publisher have made every effort to ensure the accuracy of the information herein. However, the information contained in this book is sold without warranty, either express or implied. Neither the authors and SitePoint Pty. Ltd., nor its dealers or distributors will be held liable for any damages to be caused either directly or indirectly by the instructions contained in this book, or by the software or hardware products described herein.

Trademark Notice

Rather than indicating every occurrence of a trademarked name as such, this book uses the names only in an editorial fashion and to the benefit of the trademark owner with no intention of infringement of the trademark.



Published by SitePoint Pty. Ltd.

424 Smith Street Collingwood
VIC Australia 3066.

Web: www.sitepoint.com

Email: business@sitepoint.com

ISBN 0-9752402-4-2

Printed and bound in the United States of America

About the Author

Chu Yeow is a software developer by day, a blogger by night, and a sometime freelance Web developer. He holds a Bachelor's degree in Computer Engineering from the National University of Singapore. He keeps a well-read Mozilla-related blog titled *redemption in a blog*¹, where he posts the latest Mozilla-related news and Firefox tips, and reviews blog-worthy extensions and random technical tidbits.

Chu Yeow has written several articles for SitePoint, in particular contributing Firefox tips for its community newsletter, the *Community Crier*. He is also a SitePoint Forums moderator and holds the appointment of Programming Team Advisor.

When he isn't working, this self-professed geek enjoys tinkering with his hosted Gentoo Linux box, anime, World of Warcraft (just one more level!), and good fantasy books.

About the Expert Reviewer

Asa Dotzler is best known for his work on the Mozilla project. In addition to being the QA lead, including his role as community quality advocate for the project, Dotzler is also the Firefox and Thunderbird product release manager. When he's not helping new contributors test Mozilla and report bugs, he's working to define requirements for, and ship, Mozilla product releases.

Asa maintains an active Weblog² with an emphasis on Mozilla development and news. Other interests include the exploration of Mars and other astronomy-related matters, general technology news, and his pet cat, Ptolemy, a female so-named for her feline-centric view of the universe (after Claudius Ptolemaeus and his geocentric theory).

About the Technical Editors

Nigel McFarlane is the Mozilla community's regular and irregular technical commentator. He is the author of *Firefox Hacks* (O'Reilly Media) and *Rapid Application Development with Mozilla* (Prentice Hall PTR). When not working for SitePoint, Nigel writes for a number of trade publications and for the print media. He also consults to industry and government. Nigel's background is in science and technology, and in Web-enabled telecommunications software. He resides in Melbourne, Australia.

About SitePoint

SitePoint specializes in publishing fun, practical, and easy-to-understand content for Web professionals. Visit <http://www.sitepoint.com/> to access our books, newsletters, articles and community forums.

¹ <http://blog.codefront.net/>

² <http://weblogs.mozillazine.org/asa/>

For my parents.

Table of Contents

Introduction	ix
A Little Web Browser History	x
Firefox vs. the Mozilla Application Suite	xi
Who Should Read This Book?	xi
What's In This Book?	xii
What's On The CD?	xiii
The Book's Website	xiv
Updates and Errata	xiv
Get The Rest From SitePoint	xiv
The SitePoint Forums	xiv
The SitePoint Newsletters	xiv
Your Feedback	xv
Acknowledgements	xv
 1. Introducing Your New Favorite Web Browser	1
The Case For A Better Browser	1
A Best-of-Breed Open Source Product	2
A Browser that Works for You	2
The "Right Set of Features"	8
Elimination of Annoyances	10
Security	10
Industry-Leading Standards Support	12
A Web Professional's Best Friend	13
Complacency at Microsoft	14
Getting Firefox	16
Getting the Latest Windows Release	16
Getting the Latest Mac OS X Release	18
Getting the Latest Linux Release	20
Getting the Latest Release on Other Platforms	21
Importing Settings from Other Browsers	21
Summary	23
 2. Essential Browsing Features	25
Tabbed Browsing	25
A Short History and a Warning	26
Tabs vs. Multiple Windows	27
Using Tabs	32
Search and Search Tools	39
There's More than One Way to Find it!	40
FastFind: Find As You Type	52

Non-invasive Browsing	54
Popup Blocking	55
Disabling Annoying JavaScript	58
Downloading for Dummies	61
Downloading Files	61
The Download Manager	62
Installing Plugins	66
Getting to your Email	67
Summary	68
3. Revisiting Web Pages	71
Firefox Bookmarks	72
Importing and Exporting	72
Creating a Bookmark	74
Using the Bookmarks Toolbar	76
Playing with the Bookmarks Manager	77
Spotting the Bookmarks Sidebar	79
RSS and Live Bookmarks	82
RSS Reconnaissance	82
Creating a Live Bookmark	85
Clever Use of Live Bookmarks	88
Firefox Remembers for You	88
Using History	88
Managing Cookies	90
Auto-Completing Forms	94
Using the Password Manager	95
Clearing your Tracks	99
A Brief Look at Security	100
Secure HTTPS Pages	100
Summary	102
4. Dressing Up Firefox	105
Introducing Themes	105
Shopping for Themes	106
Working with Themes	107
Installing a Theme	108
Uninstalling and Changing Themes	108
Updating Themes	109
A Tour of Successful Themes	110
Quite	110
Phoenix	110
Gray Modern	112

Saferfox Xpanded	112
Plastikfox Crystal	112
Summary	114
5. Personalizing Firefox	115
Installing Extensions	116
Taming Tabs with miniT	116
Installing from other Sites	120
Installing from a Local File	121
Using the Extension Manager	123
Updating Extensions	123
Disabling and Uninstalling Extensions	124
Fixing Problems with Safe Mode	125
Recommended Extensions	127
Saving and Recovering your Place	127
Handling Irritating Page Content	134
Adding Helpful Page Content	142
Searching in New Ways	147
Upgrading your Mouse	152
Better Downloading	157
Getting the News	165
Fun Extensions	168
CuteMenus	168
FoxyTunes	170
Summary	170
6. Tips, Tricks, and Hacks	173
The Secret Named about:config	173
Modifying Preferences	175
Enabling Resizable Frames	175
Modifying Scripted Popups	176
Single Window Mode	177
Speeding up Firefox	180
Inline URL Auto-completion	181
Identifying Firefox Profiles	182
Where, oh Where, is my Profile?	183
Files in your Profile	185
Styling the User Interface	188
Editing Profile Files with ChromEdit	188
Marking Unread Tabs	189
Shifting the Sidebar to the Right	189
Disabling Psychedelic Content	190

Making the Back Button Bigger	191
Making the Search Bar Bigger	192
Simplifying the User Interface	193
Removing the Search Bar	193
Maximizing Browser Real Estate	197
Hidden Stuff: the Calculators	199
Using the Profile Manager	200
Creating a New Profile	201
Deleting a Profile	202
Backing up a Profile	203
Restoring a Saved Profile	204
Setting Cross-Profile Preferences	205
Adding a Global <code>all.js</code> File	205
Adding a Global <code>user.js</code> File	206
Summary	207
7. Web Development Nirvana	209
Firefox's Standard Tools	210
Viewing Source Fundamentals	210
Viewing Selection Source	210
Page Info Document Reports	212
Popup Cookie Tracking	218
JavaScript Console Techniques	219
DOM Inspector Content Analysis	221
Must-Have Developer Extensions	234
Logging with LiveHTTPHeaders	235
The Web Developer Toolbar	237
Easy Validation with Checky	249
Easy Colors with ColorZilla	251
Easy FTP with FireFTP	255
Greasemonkey	257
Venkman: Advanced Debugging	260
Summary	261
8. Living On The Edge	263
News, Views, Gossip, and Facts	264
Getting the News	264
Getting Views And Opinions	265
Getting Reference Material	266
Troubleshooting Firefox	267
Problem Extensions and Themes	267
Firefox Won't Start on Windows	268

Firefox says “Profile Already in Use”	268
Firefox Profile is Corrupt	269
Strange Windows on Mac OS X	270
Uninstalling Firefox	271
Seeing Something Weird	272
Playing at the Cutting Edge	272
Defining “Nightly Builds”	273
News on Nightly Builds	275
Getting Nightly Builds	277
Mozilla’s Bugzilla	280
Jumping on the Firefox Bandwagon	281
Creating Web Magic	281
Summary	282
Index	283

Introduction

You've read about the new Firefox browser in newspapers, magazines, Websites, and favorite blogs. Perhaps you've even seen it in the news. And maybe a colleague has been raving about it at the office water cooler. Wherever you've seen it, Firefox has grabbed your attention long enough to start you wondering what the big deal is. You picked up this book, didn't you? Now you'll find out what all the fuss is about!

Firefox Secrets is not just a book about a browser's hidden secrets; it's also an explanation of how and why Firefox will make your Web surfing experience smoother and more productive. It jump-starts you on Firefox's various features and explores the rationale behind the browser's design. Scattered throughout the text are tips and little-known facts that take you even further toward an improved Web browsing experience.

Many, many people use Firefox as their primary Web browser, for myriad reasons. Switch stories abound and the Firefox community is full of enthusiasts who are as keen and passionate as Macintosh enthusiasts. It feels as if we're already deep into Browser Wars II (the sequel).

Firefox, an Open Source product that's available for free from the Mozilla Foundation, has really come into its own as a functional, standards-compliant Web browser. Conceived with a focus on usability, and offering a trim but functional user interface, Firefox was intended as a viable alternative to older Web browsers, such as Internet Explorer, and has succeeded in this goal.

Perhaps the word "secrets" in this book's title caught your attention. Indeed, tips, tricks and hacks are an integral part of this book. They'll usually appear as a relevant Firefox feature is first discussed, and then again in a chapter specially dedicated to such manipulations. Hidden preferences, profile modifications, and user interface tweaks are some of the secrets you can expect to uncover.

Beyond the browser, this book also describes Firefox extensions: little programs that you can add to Firefox to enhance its functionality to suit your needs. A whole chapter is devoted to the best and most remarkable extensions available.

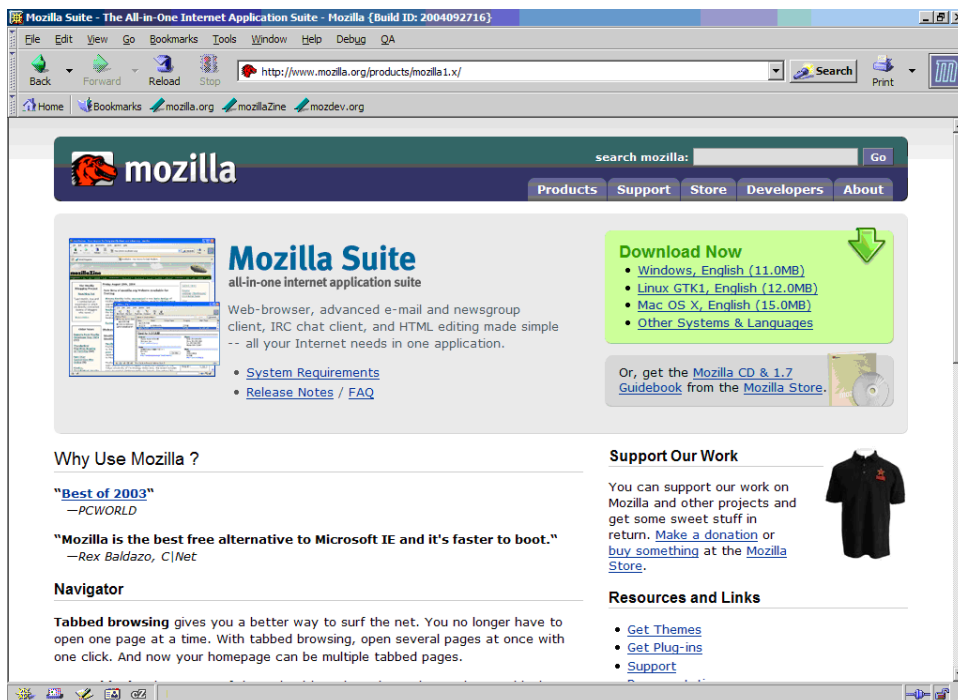
Firefox Secrets will equip you with a thorough understanding of how best to use Firefox to suit your browsing preferences, and how to enhance your productivity both at work and at play. With Firefox, you'll be able to boldly go where you've never been before. Enjoy!

A Little Web Browser History

In the beginning, there was Netscape Navigator, which begat Mozilla, which begat Mozilla Firefox.

Actually, the family lines were never that clearly drawn: the Netscape Navigator Web browser had a project mascot named “Mozilla” almost from the beginning. When Netscape Communications released the source code for Netscape Communicator (a suite of Internet tools that superseded Netscape Navigator), the Mozilla Foundation took over the development process, the Mozilla project, and the software. The first outcome of that project was a toolset that included a browser simply called “Mozilla.” Those tools are now disambiguated from the overloaded “Mozilla” word as the Mozilla Application Suite.

Figure 1. The Mozilla Application Suite’s Navigator.



The Suite, or Seamonkey, as it is internally known, consists of a Web browser (Mozilla Navigator), an email client (Mozilla Mail), an IRC (or Internet Relay

Chat) client (ChatZilla), and a HTML editor (Mozilla Composer). Figure 1 shows the Navigator part of that earlier product.

Mozilla Firefox (henceforth just Firefox) began life as an experimental modification of the Mozilla Navigator component. A small team of developers—Dave Hyatt, Blake Ross, Ian Hickson, Asa Dotzler, Bryan Ryner, and Joe Hewitt—wanted to create a browser that didn’t have the baggage that the Mozilla browser carried with it. In particular, they aimed to avoid the increasingly oversized Mozilla Application Suite, which had grown to include many features of debatable value, and instead aim for a trim but functional tool. This they have achieved with great success, as we’ll discover through the course of this book. This team has since been succeeded by Ben Goodger, who is currently Firefox lead engineer.

Firefox vs. the Mozilla Application Suite

Users new to Mozilla and Firefox are often confused about the differences between Firefox and the Mozilla Application Suite. Simply put, Mozilla Firefox is “just a browser,” whereas the Suite is an all-in-one Internet suite offering Web browsing, email, IRC, and HTML editing components.

The Mozilla Foundation’s intention is to make the Suite obsolete, replacing it with Firefox and its flagship email client, Mozilla Thunderbird. Of course, things are not as simple as they seem—there exists a large user base that’s still wedded to the earlier tool. Development of the Suite progresses apace, sometimes in parallel with, but at other times tangential to, the development of Firefox. Officially, the Suite is old-fashioned, but old-fashioned things have a way of re-inventing themselves!

The message the Mozilla Foundation is trying to convey is this: if you’re not bound by corporate policy (or personal taste) to the legacy Mozilla Application Suite, then Firefox is for you.

Who Should Read This Book?

This book is essential reading for anyone who wants to surf the Web with the best browser available!

Seriously though, this book is suited to anyone who’s curious about Firefox.

If you’re new to Firefox, this book acquaints you with the browser and goes on to teach you how to use it to improve your productivity while Web browsing.

If you're already a Firefox user, this book is especially suited to you. You'll learn how to make Firefox work most effectively for you, thanks to the wealth of tips, extensions, and even hacks contained within these pages. Expect to further improve your productivity and browsing experience with the not-so-secret secrets described herein. If there's something about Firefox that has you stumped, your questions are probably answered between these covers.

If you're a Web developer, there's an entire chapter dedicated to your needs. We'll cover the Web developer-friendly features of Firefox, and some of its more pertinent extensions. If you want to know why other Web developers rave about Firefox, this is your chance to find out. You'll never look back!

What's In This Book?

Here's what you'll find in this book.

Chapter 1: *Introducing Your New Favorite Web Browser*

This opening chapter introduces you to Firefox, and explains why it is so often plugged as a "better Web browser." You'll see for yourself whether the hype is justified. I'll also show you where to download Firefox, and how to get it installed and working well on your system.

Chapter 2: *Essential Browsing Features*

This chapter covers Firefox's major innovations, and how you can use them to make your Web browsing experience better and ultimately more productive. It teaches you how to glide across new Web territory in style! Expect to explore functionality like tabbed browsing, improved search interfaces, and features that reduce irritants such as popup windows and warnings.

Chapter 3: *Revisiting Web Pages*

This chapter covers the assistive innovations of Firefox, and explains how you can use them to manage access to all the Websites that you frequent. Here, you'll learn to use the browser to capture information about the Web just once, then place it at your fingertips, ready for return visits that don't strain your memory. Expect to delve into features such as improved bookmarks, newsfeeds, and numerous data capture tools like the Form and Password Managers.

Chapter 4: *Dressing Up Firefox*

In this short chapter, we'll see how you can easily change Firefox's look and feel with "themes." We'll also preview some of the best Firefox themes in action.

Chapter 5: *Personalizing Firefox*

In this chapter dedicated to extensions, I'll cover the extension management and review some of the best Firefox extensions available. Extensions are little programs that add functionality to the basic Firefox package. You'll learn how to install and remove extensions, and how to troubleshoot problematic extensions.

Chapter 6: *Tips, Tricks and Hacks*

Any reasonably-sized Website will contain at least a few forms through which user input can be collected. Scripting can provide some serious improvements to these forms, by way of user input validation, improved ease-of-use, the collection of more accurate feedback, and so on. Forms are built from HTML, yet the Document Object Model (or DOM) can be said to apply to forms more than it does to other Web elements, because forms have such a wide range of actions that can be manipulated through your scripts. In this chapter, we'll cover such manipulations in detail.

Chapter 7: *Web Development Nirvana*

Focused specifically on the needs of Web developers, this chapter goes into depth as we explore the development-friendly Firefox features and available extensions. Web developers who use Firefox rave about the tools and extensions that Firefox offers to help boost their productivity. Tools like the DOM Inspector and JavaScript Console are discussed, as are indispensable extensions such as the Web Developer extension.

Chapter 8: *Living on the Edge*

This last chapter gives you the details you need to start living on the cutting edge of Firefox development. You'll learn where and how to get nightly development builds of Firefox, where to get the latest scoop on Firefox news, and what to do if you get into trouble.

What's On The CD?

Included with this book is a CD-ROM that contains a treasure-trove of Firefox goodies. First and foremost, it includes the Firefox browser, but it also contains the Thunderbird email client, Firefox's sibling application. There's a collection of six themes with which you can decorate your browser, and more than twenty extensions for you to try out. As a bonus, the CD also includes sample chapters from SitePoint's other books and kits.

The Book's Website

Located at <http://www.sitepoint.com/books/firefox1/>, the Website supporting this book provides information and resources related to *Firefox Secrets*.

Updates and Errata

The errata page on the book's Website will always have the latest information about known typographical and code errors, and necessary updates for changes to technologies.

Get The Rest From SitePoint

SitePoint's main Website at <http://www.sitepoint.com/> is an intersection point for news and views on the professional Web. Take advantage of the SitePoint services that complement this book.

The SitePoint Forums

While I've made every attempt to anticipate any questions you may have, and answer them in this book, there is no way that *any* book could cover everything there is to know about Firefox. If you have a question about anything in this book, the best place to go for a quick answer is the SitePoint Forums¹—SitePoint's vibrant and knowledgeable community.

The SitePoint Newsletters

In addition to books like this one, SitePoint offers free email newsletters.

The SitePoint Tech Times covers the latest news, product releases, trends, tips, and techniques for all technical aspects of Web development. The long-running *SitePoint Tribune* is a biweekly digest of the business and moneymaking aspects of the Web. Whether you're a freelance developer looking for tips to score that dream contract, or a marketing major striving to keep abreast of changes to the major search engines, this is the newsletter for you. *The SitePoint Design View* is a monthly compilation of the best in Web design. From new CSS layout methods

¹ <http://www.sitepoint.com/forums/>

to subtle PhotoShop techniques, SitePoint's chief designer shares his years of experience in its pages.

Browse the archives or sign up to any of SitePoint's free newsletters at <http://www.sitepoint.com/newsletter/>.

Your Feedback

If you can't find your answer through the forums, or you wish to contact us for any other reason, the best place to write is books@sitepoint.com. We have a well-staffed email support system set up to track your inquiries, and if our support staff are unable to answer your question, they send it straight to me. Suggestions for improvements, as well as notices of any mistakes you may find, are especially welcome.

Acknowledgements

I'd like to thank the SitePoint team for making this book possible, and for being forbearing when deadlines slipped. Simon Mackie, in particular, had to deal with this rookie author who wasn't always able to devote his full concentration to the task at hand. It was a great surprise (and delight) to be asked to contribute to SitePoint's Firefox title, and my gratitude goes to SitePoint for giving me this opportunity.

My appreciation also goes to Asa Dotzler for being such a nice and forthcoming guy when asked to technically review a Firefox book, despite his busy schedule and workload with the Mozilla Foundation. Asa's reviews relieved me of several misconceptions, and I even picked up a few secrets from him (and yes—those secrets appear in this book).

A special thanks goes to the Firefox community: developers, evangelists, users, all of you. Firefox is the success it is today because of you, and I loved being able to be a part of it in my own small ways. It is telling that, when asked what my interests are, I mentioned Firefox first in that list.

To my blog readers: thanks for reading, or should I say “not reading.” Several visitors complained about the lack of activity on my blog while I was writing this book. Well, your support means a lot to me, and I'll probably be back to my old level of blogging activity by the time you read this.

2

Essential Browsing Features

If you feel that the Web has lost its sparkle, that's probably because you're slogging across it in an old browser. I'm over here on the other side, and the grass is not only greener: there are none of those microscopic grass bugs that cause nasty rashes!

This chapter is all about moving forward to browse the Web with a new sense of style. We'll investigate the major features of Firefox and I'll show you how to use them. You'll learn how to explore the Web more efficiently, and with fewer interruptions. You'll also see how you might (if you so choose) let go of a few old Web surfing habits that may be holding you back; window management habits and Web search habits are two very important examples. There may even be pictures of attractive models, both male and female. Okay, there are no models. Let's explore Firefox!

Tabbed Browsing

Tabbed browsing has been called “the best thing since sliced bread and the biggest fundamental improvement in Web browsing in years.”¹ It has also been criticized, rejected and labeled a useless non-innovation. You must be wondering what the

¹Walter S. Mossberg, Staff Reporter of The Wall Street Journal
http://www.wsjclassroomedition.com/archive/04mar/ptec_browser.htm

big deal is with tabbed browsing and which point of view is right. You might also be curious about which side of the argument I, your intrepid author who pretends to know what's best for you, am on. You'll find out soon enough!

If you're a stranger to tabbed browsing, you're probably trapped in a somewhat old-fashioned Web surfing pattern. Perhaps you click on an interesting link, wait for that link to load, then press the Back button when you've finished reading that new page. Alternatively, perhaps you right-click (or context-click on Mac OS X) on a hyperlink and select Open in New Window on those occasions when you don't have time to wait for the new page to load.

Either way, prepare to be blown away when you see how effectively tabbed browsing eliminates waiting and unnecessary backwards and forwards tracking. You'll be wondering how you could ever have been satisfied browsing one Web page at a time in a single window. Those who have already made the leap may still need some convincing as to why tabbed browsing is better than browsing with multiple windows. I'm here to encourage you to consider the benefits.

Let's learn everything about use of tabbed browsing in Firefox. But remember: you are not forced to use it. If you're more comfortable with a single window per Web page, then that's your choice. Don't deny yourself the possibility of a better alternative, though.

A Short History and a Warning

Tabbed browsing is not that new an idea; in fact, it wasn't invented by Firefox. Probably NetCaptor,² a third-party program that provides an alternative tabbed browsing interface for Internet Explorer, pioneered this approach. The Mozilla Application Suite followed hot on the heels of that tool, and more recently, Opera² has, too. Firefox, being a derivative of the Mozilla Application Suite, naturally inherited its tabbed browsing capability. Of course, spreadsheet tools like Microsoft Excel have used tabs in the worksheet display area for a long time.

Currently, most graphical browsers support tabbed browsing. A quick (and incomplete) list of tab-enabled browsers reads: Firefox, Opera, Mozilla, Safari, Konqueror, Netscape, OmniWeb, and Camino. This list doesn't include programs like NetCaptor, Maxthon, Avant Browser, and Crazy Browser, which add tabbed

² http://en.wikipedia.org/wiki/Tabbed_browsing

² Opera developed a Multiple Document Interface (MDI), which is not the same as the Tabbed Document Interface (TDI) of tabbed browsing; Opera has added a faux-tabbed browsing interface in Opera 7.60 preview 3 (and later).

browsing functionality to Internet Explorer. The only notable exception is Internet Explorer itself, which may possibly be feature-enhanced with tabs in version 7.0.

It's important to realize that the add-ons that provide tabbed browsing functionality in Internet Explorer (such as the aforementioned NetCaptor, Maxthon, and kin) carry the same faults as the Internet Explorer engine, since they are no more than superficial "skins" over this product. This means that, when you use one of these variant browsers, you remain open to the set of security risks implied by Internet Explorer.

Tabs vs. Multiple Windows

Here I am, stepping onto my soapbox and making the case for tabbed browsing against the alternative: multiple browser windows. Naturally, I'll use Internet Explorer as the prime example for the case of multiple browser windows, since it's the most popular tab-free browser. I'll make objective arguments (I hope!) for both tabbed browsing and multiple windows, and leave it to you to make the decision as to which option suits you best.

Organizational Differences

Tabbed browsing doesn't prevent the use of multiple windows; in fact, you can opt to use a browser that supports tabbed browsing in the same way as you use Internet Explorer (i.e. displaying a single window per Web page). You can use this compatibility to your advantage, mixing and matching both styles to suit your needs. Tabbed browsing groups related tabs in a single window, so you can have one window of tabs for work, another window for email and personal stuff, and so on. Tabbed browsing also lets you keep related tabs in a logical group. Multiple windows, on the other hand, are just that: a single window displays a single Web page.

There's not much I can say to help the old-style multiple window case, even though I want to be objective. One small advantage is that closing a non-tabbed window dismisses only the single page it displays. Closing a tabbed window can dismiss a whole set of pages, some of which you may have meant to keep.

Managing Multiple Web Pages

It's second nature for most of us to work with multiple windows simultaneously. Naturally, we expect that those windows can be operated in a manageable way, but the crunch really comes when we have a large number of Web pages open.

Does Figure 2.1 below, which shows the Windows taskbar, look manageable to you? There are 14 browser windows open!

Figure 2.1. Using multiple browser windows in IE.



I know I'd have a hard time finding the Web page I wanted here, particularly as each taskbar button shows the same unhelpful Internet Explorer icon. One solution, of course, is to use the “Group similar taskbar buttons” option that’s available on Windows XP; Linux KDE and Gnome users have an equivalent feature. Grouping helps—at least you can see the page titles, as shown in Figure 2.2. Personally, though, I dislike this feature because it requires me to click twice to reach the particular window I want.

Figure 2.2. Grouping IE windows makes browsing slightly easier.



In my view, tabs are a better solution. Figure 2.3 shows the same Web pages opened in tabs in Firefox.

Tabbed browsing suffers the same page title truncation problem as the taskbar, although a few more letters show in the given tab space. But, with the same number of windows open, it’s still hard to work out which page is the one I want. At least the tabs display helpful icons (called favicons) for Websites that provide them. You can see that about half of the tabs displayed in Figure 2.3 have favicons.

Figure 2.3. Multiple tabs displaying in Firefox.

Another redeeming feature of tabbed browsing is that this system allows you to juggle more Web pages than would be possible using multiple separate windows. You can handle more Web pages because there's no competition for space on the tab bar (the section of the display that holds the tab names and icons). Multiple window browsing requires each IE window to compete with other open programs for space on the taskbar. That includes non-IE windows, the system notification area at the bottom right, the Start bar, the Quick Launch bar, and any other bars you may happen to have configured.

Which solution stands out? Well, both have merits. Firefox helpfully displays an indicative icon (a favicon) for different Websites, which, to my mind, is much more helpful than the showing the same "e" icon for every page. These displays vary a little between the Macintosh and Linux desktop environments, but I'm sure you get the picture.

Avoiding Context Pollution

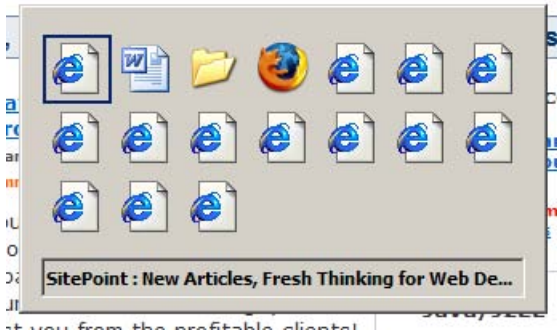
There is a little issue I like to call "context-switching pollution." Application context-switching occurs when you switch from one window to another³ on the desktop. In Windows, you do that by clicking on a taskbar item, or by pressing **Alt-Tab** to bring up another window. Mac OS X users are probably big fans of Exposé. In Linux, depending on your Desktop Environment, you may have Windows-like **Alt-Tab** context-switching, or even an Exposé-like feature. When you context-switch, Windows displays an icon (a context object) for each available window; you then pick the one you want.

Now, if you have tabbed browsing in place, you'll probably have a few windows open at most. In fact, most of the Firefox users I know have no need for more than a single window to hold all their tabs. I usually stick to a single window, myself. You can open as many tabs as you want within a window, and the bonus

³My apologies to techies for this simplification and my abuse of the term "context-switching."

you receive is this: you have just one context object for that set of tabs. One window: one context object. By comparison, imagine that you have multiple IE windows open: one for each Web page. Each of these windows counts as a context object. And you have a bad case of context-switching pollution, my friend! Figure 2.4 shows what context-switching pollution looks like.

Figure 2.4. Context-switching pollution in Windows XP.



How are we ever going to quickly find the window we want without some furious Windows-style **Alt-Tabbing**? Figure 2.5 shows a similarly cluttered summary delivered by Exposé on Mac OS X.

Although Exposé fares better than the Windows context-switcher, you’re still left to perform laborious squint-and-click selection.

Tabbed browsing reduces those nasty multiple windows into individual tabs in a *single* window—two or three windows at most. Bye-bye, context-switching clutter!

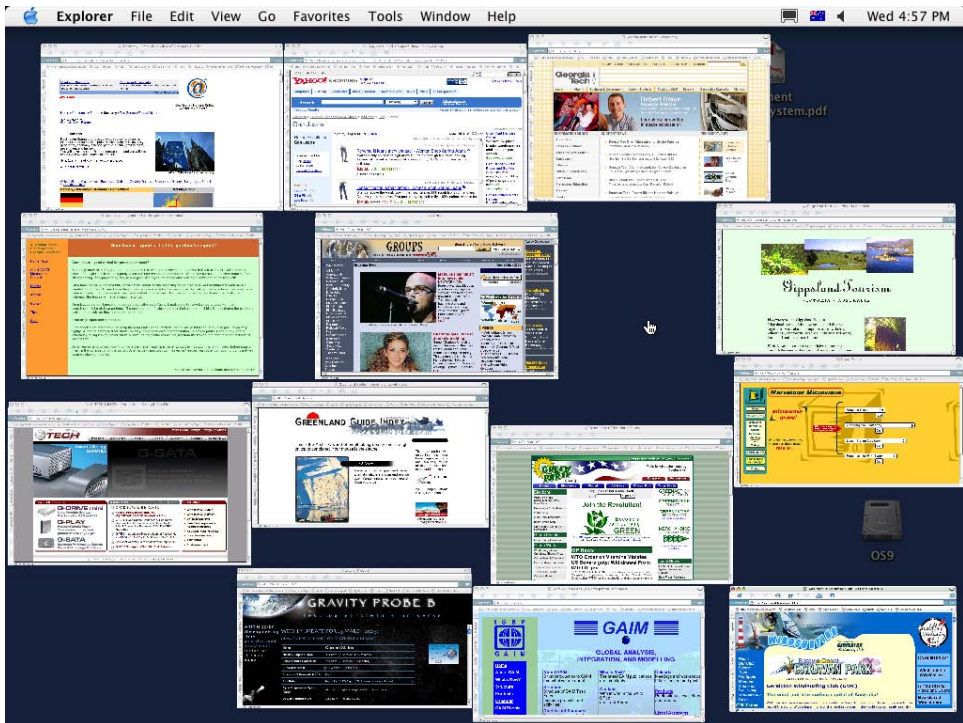
Productivity Differences

When you read a Web page in a tabbed browser, you can open interesting links in “background” tabs, which will load those pages as you continue to read the original. You can access those new pages in your own time, once you’ve finished with the current page. By then, they will likely have finished loading, and will be ready to view.

When you use a single window per Web page, you have to open interesting links in new windows to achieve the same effect. In Internet Explorer, this will open a new window, which then has the “focus:” it’s brought to the front of all other

windows. This stealing of focus may or may not be what you want. If it isn't, too bad, you just have to navigate back to the original window again. You can't change this behavior in IE; in Opera, at least, there is an 'Open behind' option that keeps the focus on the current page. In a tabbed system, you're able to tell the browser whether or not to open new tabs in the background, and when you want them to steal the focus. We'll see how to do both later.

Figure 2.5. Exposé pollution on Mac OS X.



I've found opening links in background tabs to be the most productive and efficient method of Web browsing. There is a near-perfect match between the basic concepts of Web browsing, and of opening interesting links in background tabs. Think about it: Web pages are built on hypertext, which means that there are always hyperlinks to other Web pages. Some links may interest you; most probably won't. You'll want to visit those links you do care about, so you click on them. But, wait! We're talking about "links," not "a link." Usually, more than a single link will catch your attention (especially when using Google or one of the other search engines). You'll want to visit all those interesting links, and it's a natural

process to open those links as you read the current page, so that you aren't distracted from the current task, and visit them when you're done. Tabbed browsing allows you to do that. IE doesn't, because it steals focus when you open links in a new window.

The Verdict

Needless to say, tabbed browsing makes a very strong case for itself. I loved tabbed browsing the first time I tried it (in the Mozilla Application Suite). To me, it feels like the way the Web was meant to be browsed!

There is more to tabbed browsing than tabs alone, as we'll see as we review the other features of Firefox. Firefox is built around the tabbed browsing paradigm, and, throughout this discussion, we'll explore instances of how this integration can be used to further improve your browsing experience.

Using Tabs

By now, you're probably convinced of the superiority of tabbed browsing—or are, at least, willing to give it a try! In this section, we'll see how to use tabbed browsing in Firefox.

Opening, Closing and Changing Tabs

Opening a new tab is easy; it can be done in many ways.

The simplest way to open a new tab is the middle-click. Middle-clicking (or scroll wheel-clicking) on a link creates a new tab and loads the linked page into it. This is also the most efficient way to open links, so it's worthwhile to learn to middle-click if you aren't used to doing so. Most people aren't even aware that scroll wheels are clickable, but you can simply click the wheel as if it were a button, as the wheel has both button and scrolling functions. If you have a three-button mouse, just click away. If you have a two-button mouse with a scroll wheel, middle-clicking can be a bit disconcerting at first. On Mac OS X, use command-left-click instead. If you like middle-clicking, you can make this a standard on the Macintosh by reassigning command-click to the mouse wheel (middle button) via the Mac OS X mouse driver.

Note that if you miss the link with the mouse pointer when middle-clicking, Firefox goes into “free scroll” mode, which is probably not what you're after. If

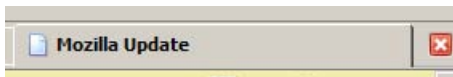
that happens, click again while the mouse pointer is still away from any links, and you'll be back to normal.

Here are a few other ways to open tabs:

- ☐ **Ctrl-T** opens a new, empty tab.
- ☐ Right-click (or context click on the Mac) a link, and select Open Link in a New Tab from the resulting context menu.
- ☐ Hold **Ctrl** down while clicking a link.
- ☐ Type a URL into the location bar, then hit **Alt-Enter**. This loads the URL in a new foreground tab.
- ☐ Press **Ctrl-Enter** when a link is selected (for example, after you've run a page search for it and the link is highlighted).
- ☐ Drag a URL (from a Web page or an external application) and drop it onto the tab bar.

Closing tabs is also easy and flexible: simply middle-click on a tab's description (the bit that sticks up above the tab content). You can also use **Ctrl-W** to close the current tab, or you can right-click on the tab and select Close Tab. Choose whichever groove suits you. Finally, you can click on the Close Tab icon, which appears as an "X" on the right-hand end of the tab bar, and "Poof!"—no more tab. Figure 2.6 shows the Close Tab icon as it appears on the tab bar.

Figure 2.6. The Close Tab icon adjacent to the tab bar.

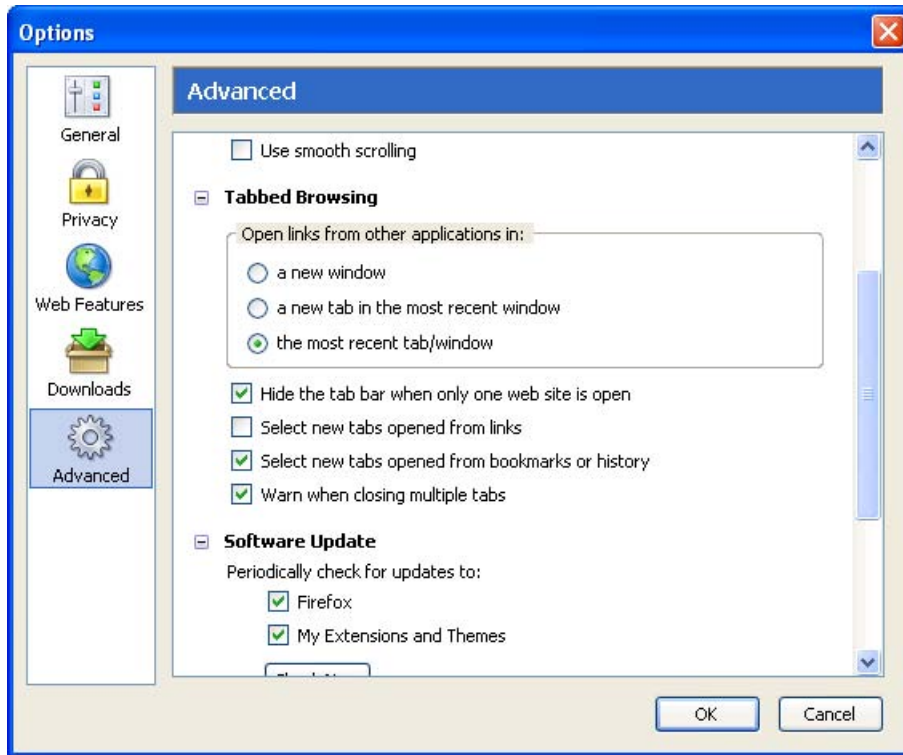


Once you've got your tabs set up, switching focus between them is easily done: just click on the required tab. Less obvious ways to switch tabs are to use **Ctrl-Tab** to switch focus to the next tab (to the right of the current tab), and **Ctrl-Shift-Tab** to access the previous tab (to the left of the current tab). This is very useful if you've got your hands on the keyboard most of the time.

Tab-Related Preferences

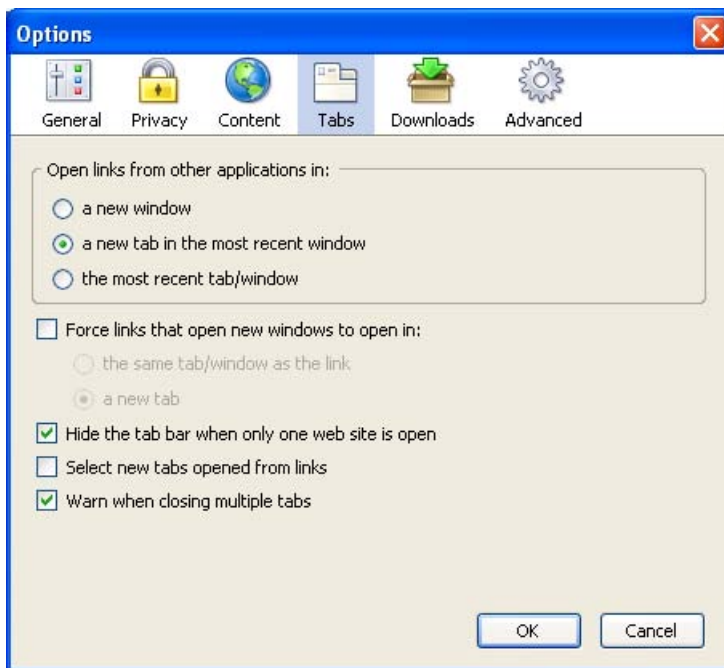
Let's take a look at the available user preferences that are related to tabs and tabbed browsing. Start Firefox and choose Tools > Options (Windows), Firefox > Options (Macintosh) or Edit > Preferences (Linux) from the menu bar, then select the Advanced panel. You should see a dialog box that's something like the one shown in Figure 2.7.

Figure 2.7. Tabbed browsing preferences in Firefox 1.0.



Note that the options panel has been redesigned slightly for Firefox 1.1 and beyond: there, you must click a tab rather than an icon to see this information. Figure 2.8 shows the shape of things to come.

Figure 2.8. Tabbed browsing preferences in Firefox 1.1 and beyond.



You can see that, although there's a little reorganization, the options are much the same in both versions. Here's what each preference does:

Open links from other applications in:

If you click on a link displayed in another program, such as Microsoft Word or Adobe Acrobat Reader, this preference dictates how that link will be displayed by Firefox. I prefer to have links from other applications open in a new tab in the most recent window, because I usually use a single window, and I don't want to lose the Web page on my most recent tab. Of course, links will only load in Firefox if Firefox is set as your default browser.

Force links that open new windows to open in:

This is a newish feature that can be used to tame popup windows: check the box to turn it on. Instead of popups launching in a separate, new window, or being blocked by the popup blocker (discussed later), this option lets that new content be created inside an existing tab or an existing window. This option is available only in Firefox versions 1.1 and later.

Hide the tab bar when only one Website is open

If checked, this hides the tab bar when only one Web page is open. Personally, I have this unchecked (it's checked by default), because it's disconcerting to have the tab bar disappear and re-appear. This option was made the default as a deliberate design decision to enable users who don't use tabbed browsing to enjoy more screen real estate. Users of tabbed browsing, however, will probably prefer to have this option unchecked.

Select new tabs opened from links

This is the preference I was talking about earlier, which allows you to choose whether new tabs open in the background, or steal focus from the current page. Checking this option will cause links opened in new tabs to steal focus. I recommend that you leave this unchecked to gain the productivity benefits I mentioned earlier.

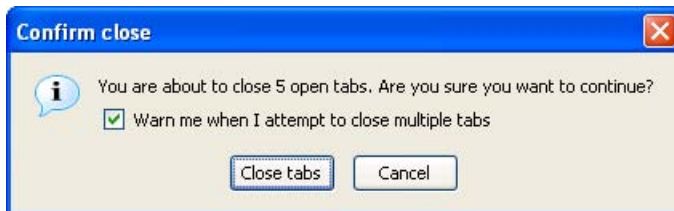


Those who have the Select new tabs opened from links preference set one way or the other can reverse their setting temporarily with the **Shift** key modifier. For example, if you've set Firefox up so that new tabs opened from links load in the background, but you want a particular link to open in the foreground, just hold on to the **Shift** key when clicking the link (i.e. **Shift**-middle-click, or **Ctrl-Shift**-left-click).

Warn when closing multiple tabs

This option will display a warning dialog if you take an action to close more than one tab simultaneously, as will occur if you try to close a window in which multiple tabs are open. This is a useful warning that can prevent you from accidentally closing a window full of tabs, but it could be annoying if you mean to close that window. Figure 2.9 shows this warning.

Figure 2.9. Warning dialog shown when closing multiple tabs simultaneously.

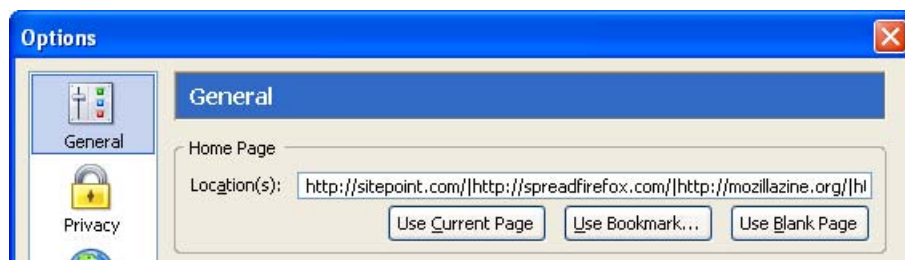


As you can see, you don't have to change any of the options if you are a "normal" user: Firefox is built so that sensible configuration choices are available from the start.

A Single Homepage: So 1999!

Another good thing about tabbed browsing is that tabbed browsers are no longer stuck with a single homepage. If you're like most other Web surfers, you probably have more than one favorite Web page: pages that you absolutely *must* visit the first time you launch your Web browser. I, for example, have a total of eight URLs that I want to check at the start of every Web surfing session.⁴ What's a user of a Single Document Interface (SDI)⁵ Web browser—Internet Explorer, for example—to do? Well, they'd better start looking through their bookmarks! Those lucky users of tabbed browsers, on the other hand, can set multiple homepages and load all of their favorite pages at once. Figure 2.10 illustrates the format required for setting multiple homepages via the Options dialog box.

Figure 2.10. Specifying a tabbed set of homepages.



Note the vertical bar (|) that separates each URL: one URL is shown per tab. You simply load the preferred Web pages into tabs, and mark the set as your homepage. For example, if you want <http://www.sitepoint.com/> and <http://www.spreadfirefox.com/> to be your homepages, load them up in Firefox, go to Tools > Options > General (or Firefox > Preferences > General on Mac OS X, or Edit > Preferences > General on Linux), then click the Use Current Pages button.

Alternatively, you can hand-enter the URLs you want to set as your homepages in the Location(s) text box, separating each using the pipe character (|) as shown above. For example, if you want sitepoint.com and [Spreadfirefox.com](http://spreadfirefox.com) to be your

⁴In the unlikely event that you are interested, they are: <http://animesuki.com/>, <http://www.sitepoint.com/forums/usercp.php>, <http://bloglines.com/myblogs>, <http://forums.mozillazine.org/>, <http://gmail.google.com/gmail>, <http://blog.codefront.net/>, <http://spreadfirefox.com/>, <http://www.worldofwarcraft.com/>

⁵A Single Document Interface (SDI) is, simply put, a way to organize applications into individual windows, one for each application (or each instance of an application). If you've used Internet Explorer, you already know what an SDI is.

homepages, enter **http://sitepoint.com/|http://spreadfirefox.com/** into the Location(s) textbox.⁶

A third approach is to click on the Use Bookmark... button, and then select the bookmark folder that contains the bookmarks you want to use as your homepages. We'll cover bookmarks in more detail later.

Setting more than a single homepage does have a disadvantage, though. If all you want to do is a quick Web search, Firefox startup is too clever (or perhaps too dumb), and insists upon loading your entire set of homepages instead. This does slow things down a bit, especially on a dial-up connection, because loading eight Web pages is almost certainly slower than loading just one. You can always press **Ctrl-T** to get a new tab and turn eight into nine, if you're feeling impatient. You can then type your desired URL into the ninth tab and have it load at (hopefully) more than one-ninth of its normal rate.

Despite the performance implications, it's hard to deny the convenience of loading up multiple Web pages at once, and it's even harder not to feel sorry for those poor Internet Explorer users still facing dilemmas like, "Do I want Yahoo! email as my homepage, or should I set it to sitepoint.com instead?" With Firefox, you can avoid such decisions. Just set all your favorite Web pages as your homepage and be done with it!

Back, Forward and Home Buttons

Now, Firefox is a tabbed browsing Web browser, right? Of course it is. So it would make sense if Firefox could open the previous page in a new tab if you middle-click on the Back button, right? In fact, this is exactly how Firefox behaves.

You can middle-click (command-left-click on Mac OS X) on all of the Back, Forward, and Home buttons in the toolbar to open the corresponding link(s) in a new tab (or tabs). If your homepage is a set of four pages in separate tabs, middle-clicking on the Home button will add four tabs to your current tab set. Figure 2.11 reminds us of what these major buttons look like. There's no special tab action for the Reload or Stop buttons.

⁶The pipe character key is located near the **Backspace** key on most PC keyboards, and is represented by two vertical dashes placed on atop the other.

Figure 2.11. Firefox’s toolbar icons.



The provision of tab support within the main browser buttons clearly demonstrates that tabbed browsing is part and parcel of Firefox’s design. It’s neither an after-thought nor a bolted-on feature.

Search and Search Tools

Search: you can’t escape it as a fundamental Web surfing activity. Well, perhaps you can if you have the uncanny ability to guess URLs that give you the exact Web page you wanted. A good dab of clairvoyance is probably necessary as well! “Hmm... Let’s see. Are there any good Firefox ebooks online? I shall use my Powers of Uncanny Deduction to find out...” (Supernatural pause.) “Ah, I’ll go to <http://firefox-book.com/>.”

Does that even sound remotely possible? Well, if there was a Website at <http://firefox-book.com/> (there isn’t), it would probably have something to do with a book on Firefox. Relying on your powers of deduction is, in the end, unlikely to produce results, and besides, Uncanny Deduction is tremendously exhausting!

This rigmarole underscores the reality that Search is an integral part of using the World Wide Web. You can’t find the information you want using hit-and-miss techniques. Instead, we use search tools that offer systematic solutions: search engines, directories, and similar online services.

Examples of search are everywhere. Are you looking for the scoop on your favorite band? Perhaps you’re trying to learn Java programming? Google is your best bet. Are you reading up on the latest in flux capacitors and time travel for your thesis? Run a search over at ScienceDirect⁸ or arXiv.org.⁹ Looking for a good computer book, self-improvement manual, or Dragonlance⁷ novel? Amazon.com¹¹ has a nice search feature that yields customer reviews and recommendations.

If search is an integral part of the Web, it follows that it should also be an integral part of your Web browser. Firefox promises unobtrusive and integrated search

⁸ <http://www.sciencedirect.com/>

⁹ <http://www.arxiv.org/>

⁷ I’m a Dragonlance fan, in case you’re wondering. I’m not into self-improvement books, though. (It’s more fun to write your own D&D campaign settings—Ed.)

¹¹ <http://www.amazon.com/>

functionality. The Firefox developers know what you want, and they give it to you. I wish all software projects did that!

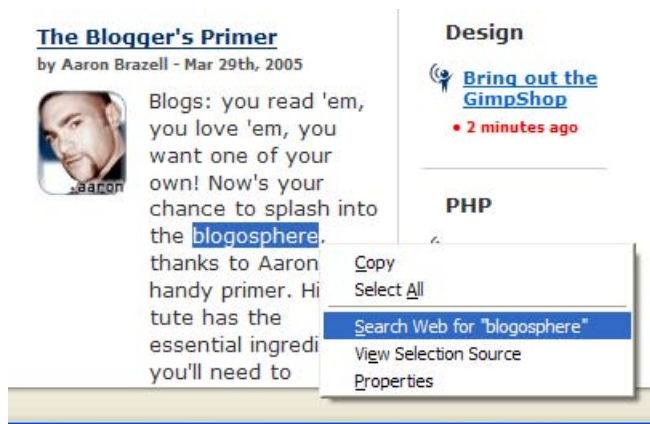
Here, we take a detailed look at Firefox's search-related features and see how you can best employ them to make things easy for yourself.

There's More than One Way to Find it!

That's right: There Is More Than One Way To Find It (TIMTOWTFI). Search is an integrated feature in Firefox, but it's integrated in several ways. There are many separate search features and several different starting points from which you can perform a search. Each method of searching is convenient for its particular situation.

For example, you can search for the meaning of a word or phrase (say, “blogosphere”) on a Web page by highlighting it, then context-clicking and selecting the context menu entry Search Web for “blogosphere.” Figure 2.12 shows this technique at work.

Figure 2.12. Context Web searching for highlighted text.

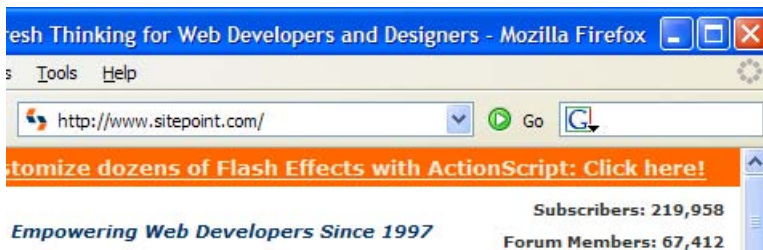


Alternately, you can surf to the search engine's homepage and type your search terms into the form (not that this is a Firefox feature, but it is a way to perform a search). Or you can... but let's put an end to these trivial examples and dive straight into discussing the more effective and efficient ways that you can search using Firefox.

The Search Bar: Drunk on Search!

When you first launched Firefox, you probably noticed that there were two textboxes into which you could type text. One of these is the location bar, into which you can type URLs; the other has a funny looking “G” next to it. Figure 2.13 shows this corner of the browser.

Figure 2.13. Search bar on the right, location bar on the left.



The “G” is actually an icon for the Google search engine, and this textbox is called the “search bar.” If you type in a search phrase, and hit Enter, Firefox will take you to the Google search results page for that phrase. Go ahead and try it out for yourself: it’s a very convenient arrangement. No longer do you have to type in `http://www.google.com/`, wait for it to load, and then search using the standard form. The search bar saves a good three seconds each time you do a search—possibly more if you’re on dial-up. Since the average Web surfer performs 12 searches per hour,⁸ that’s a good 36 seconds saved for every hour spent online, or 14.4 minutes per day, or 3.65 days per year!⁹ Imagine what you could do with those 3.65 days. You could take a three-day vacation and *still* have 0.65 days left in which to do nothing. When the Firefox folks say that Firefox helps make your online experience more productive, they aren’t just boosting the marketing hype: the productivity increases are real.

Using the Search Bar like a Pro

The search bar is easy to use, but you can do a lot more with it than merely search Google. Learn these other tricks before you embarrass yourself in front of your Firefox evangelist friends!

⁸This figure was derived through a complex process called guessing!

⁹Try doing a search on Google for “36 seconds per hour in days per year.” Google’s calculator kicks in and does the conversion for you! I knew Google’s calculator function was really smart, but *this* smart?!

To start with, hitting **Enter** loads the search results page in the current tab. At times, it may be more convenient to load in a new tab. For cases like this, hit **Alt-Enter** instead of just **Enter**.

The shortcut keys that place the input cursor in the search bar are **Ctrl-K** and **Ctrl-E**—you can use either. If, like me, you don't like to take your hands off the keyboard, those shortcuts are very useful. The way I search is to hit **Ctrl-E**, type in my query, and then hit **Alt-Enter** to specify a new tab.

You can also search using drag and drop. Simply select a piece of text (from any drag and drop-capable program, not just Firefox), and drag it into the Firefox search bar. And, if you're thinking of dragging and dropping text from a Firefox Web page specifically, there is an Even Better Way! Let me keep you in suspense for now: we'll come back to this shortly.

Adding More Search Engines

You aren't restricted to searching on Google, even if it is the best search engine around and complemented by totally cool software. You can easily add other search engines, like AltaVista, AllTheWeb, and even the new kid on the block, Clusty. To do so, click on the G icon. You should see a drop-down like the one shown in Figure 2.14, which lists the search plugins installed by default in the Firefox product.

To search with a different search engine—say, Amazon.com—simply select that item from the drop-down. The icon to the left of the search box will change from the G icon to its Amazon equivalent. Now, any searches you perform through the search bar will be run on Amazon.com (try searching for “Firefox”).

Naturally, everyone has his or her own specific needs and favorite search engines. You'll probably be wondering how to add a new option to the search bar, and you might have guessed that you simply click the Add Engines item in the drop-down to do so. Choosing that item takes you to Firefox Central, where you can add other search plugins. Figure 2.15 shows a slice of that page.

Figure 2.14. Adding and changing search engines.

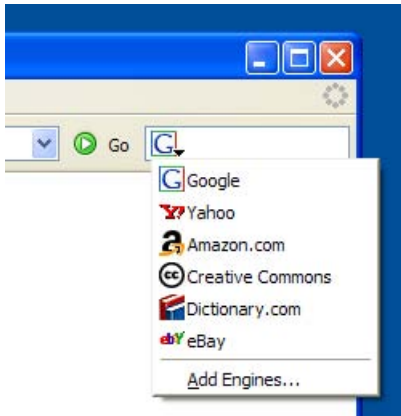


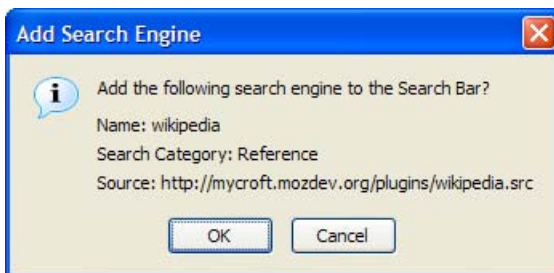
Figure 2.15. Adding new search engines at Firefox Central.

Add New Search Engines

Up in the top right corner of Firefox, there's a handy search box that puts search engines at your fingertips. When you download Firefox, a number of searches (like Google, Yahoo, and Ebay) are included but you can easily add more. There are literally hundreds of search engines available. The following are some of the most popular:

- [A9 Search](#) - Amazon's web and shopping search.
- [AltaVista](#) - Another popular way to search the web.
- [Ask Jeeves](#) - Ask it questions and get answers.

Figure 2.16. Adding the Wikipedia search plugin.



To add a particular search engine, just click on the appropriate link. Let's add the search plugin for Wikipedia (a very useful and comprehensive online encyclopedia). You should see the confirmation dialog shown in Figure 2.16 after you click on the Wikipedia search plugin link.

Click OK, and you're done! You can now see Wikipedia in the search bar dropdown. Simply select it to search Wikipedia, as shown in Figure 2.17.

Figure 2.17. Successful installation of the Wikipedia search option.



If you want to add a search engine that isn't on the provided list, go to the Mycroft project Website,¹⁴ which offers a collection of search plugins for many sites. Search for the plugin you want, or browse the search plugins categories. Over a thousand search plugins are listed here, so there's definitely something for everyone.

Removing a search engine from the search bar is a little bit of a hassle. First, use the desktop file manager (Explorer or Finder) to locate the `searchplugins` directory inside the Firefox installation directory. It's at `C:\Program Files\Mozilla Firefox\searchplugins` on Windows systems. There, two files are stored for each search plugin. One is a `.src` file, the other is a graphic, usually either a `.png` or `.gif` file. Delete the pair of files whose names match the search option that you no longer want. For example, if I wanted to remove the Yahoo! search plugin, I would simply delete `yahoo.src` and `yahoo.gif`, then restart Firefox. An interface for removing unwanted search engines is planned for inclusion in a future release of Firefox.¹⁰

Next, we'll take a look at a clever and convenient way of searching.

¹⁴ <http://mycroft.mozdev.org/download.html>

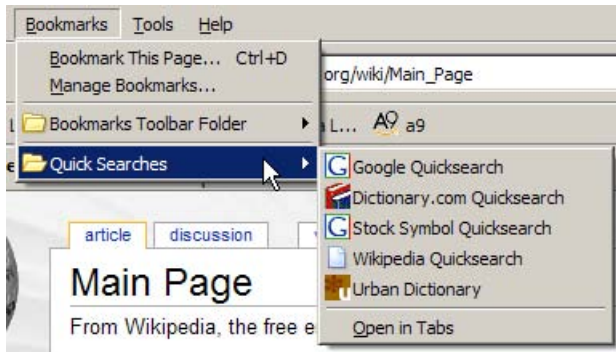
¹⁰ The related bug note is available at https://bugzilla.mozilla.org/show_bug.cgi?id=232272.

Smart Keywords

Smart Keywords are a great way to perform search queries from the location bar. With smart keywords, you can type **dict extemporaneous** into the location bar, hit **Enter**, and be taken to the Dictionary.com definition of “extemporaneous.” The hard way to do this is to go to Dictionary.com, wait for it to load, click on its search box and enter **extemporaneous**. Bah! That takes too many steps.

Firefox comes with several Smart Keywords automatically installed. You can find them in the Quick Searches folder in your bookmarks, as illustrated in Figure 2.18. You don't have to use this menu at all—it's just an easy way to see which Smart Keywords are installed.

Figure 2.18. Firefox's pre-defined Smart Keywords offering.



The predefined Smart Keywords are:

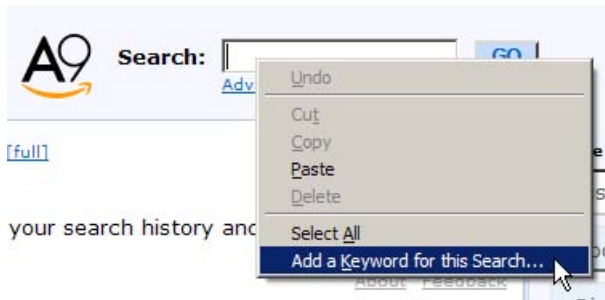
Dictionary.com	Type dict word to lookup the definition of a word.
Google	Type google search keyword(s) to perform a normal Google search.
Stock Symbol	Type quote symbol to look up stock quotes for a stock symbol. This content is sourced from Yahoo! Finance by Google.
Wikipedia	Type wp anything to access the Wikipedia entry on just about anything (Wikipedia is a free online encyclopedia).

Urban Dictionary Type **slang slang word** to look up the meaning of urban slang words, and keep up with the times!

I'm sure you're wondering how Smart Keywords are created. After all, it would be nice to construct additional keywords for searches that Firefox has not already provided for you. There are two ways to create a Smart Keyword: the easy way, and the hard way. The easy way is preferred, but when it occasionally fails, you'll have to fall back on the hard way. Let's look at both, starting with the easy way.

Let's add a Smart Keyword for A9, Amazon's relatively new search engine. First display A9's homepage.¹⁶ Right-click in the search box, and you should see the context menu item Add a Keyword for this Search..., as shown in Figure 2.19.

Figure 2.19. Adding a Smart Keyword from a search field.



The last item on the menu looks like the one we're after, and indeed it is. When you pick that item, an Add Bookmark dialog will appear, prompting you to fill in the name of the bookmark as well as the keyword. Let's name it **A9 keyword** and use **a9** for the special keyword. Save the result in a bookmark folder of your choice—I recommend you save it in the Quick Searches folder with all the other predefined Smart Keywords.

When you're done, you can try it out straight away. Enter **a9 firefox**, and you'll be taken to the A9 search results for the "firefox" search term that you entered. Great! We're making progress.

Now, let's try doing this same task the hard way. We're not doing this just so we can say we've done it to impress fellow guests at dinner parties. We're doing it because the easy way doesn't work.

¹⁶ <http://www.a9.com/>

For example, go to the Yahoo! search engine,¹⁷ and try to add a Smart Keyword the easy way. Nothing happens: no Add Bookmark dialog appears—at least, not as this book goes to print. (Technically, this is because the Yahoo! Search text field is not specified with `id="search"`).

Perhaps Yahoo! will fix this someday. Perhaps it will already have been fixed by the time this book lands in your hands: such is the nature of the Internet. Fixed or not, it's clear that many Websites do not support the easy addition of Smart Keywords.

To add a Smart Keyword the hard way, you must first study Web addresses a little. Type a query into the search box, hit **Enter**, and look at the URL that's generated. We need to pull that URL apart for our own purposes. If we use the Yahoo! Search search engine to search for "firefox," the resulting search URL looks like this: `http://search.yahoo.com/search?ei=UTF-8&fr=sfp&p=firefox`¹¹

Replace the search query ("firefox") in that URL with `%s` to create a URL that's suitable for a Smart Bookmark. A Smart Bookmark is a special bookmark that is used to define a Smart Keyword. Firefox recognizes "`%s`" as a placeholder for your keyword, and replaces it with your typed information whenever you perform a Smart Keyword search. If you enter two or more words, possibly separated by spaces, Firefox treats them all as a single keyword string. So you can only use "`%s`" once.

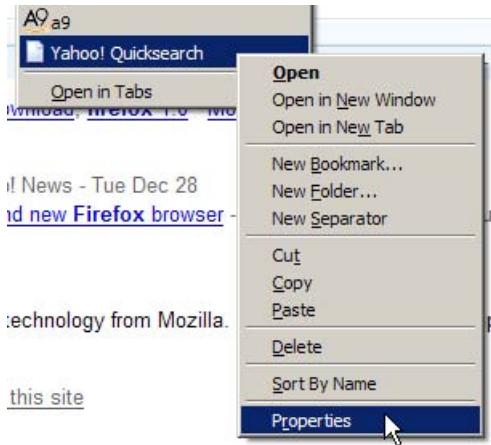
To create the Smart Bookmark, we bookmark the search page (the ordinary search page for the search engine in question), then modify the new bookmark. We change the bookmark's name to something better, like "Yahoo! QuickSearch," and tweak the bookmark so that it acts in a Smart Keyword-like manner. We then save the bookmark in the QuickSearch bookmark folder.

To do all this, find the new bookmark in the Bookmarks menu, select it, right-click, and select Properties. Figure 2.20 shows the menu that holds the Properties item.

¹⁷ <http://search.yahoo.com/>

¹¹ Those who are familiar with URL query strings, and understand how certain search engines work, will recognize that only `http://search.yahoo.com/search?p=firefox` is necessary to perform the search. The rest is extra information for Yahoo!, like the character encoding for your search query and where you conducted the search.

Figure 2.20. Displaying a plain bookmark's properties.



Choosing Properties should bring up the bookmark properties dialog, in which we can make all the required edits. Update the name, replacing “firefox” with the special **%s** placeholder, and provide a keyword (I chose “y” for “yahoo”). You can also add a description if you wish. Figure 2.21 shows these changes.

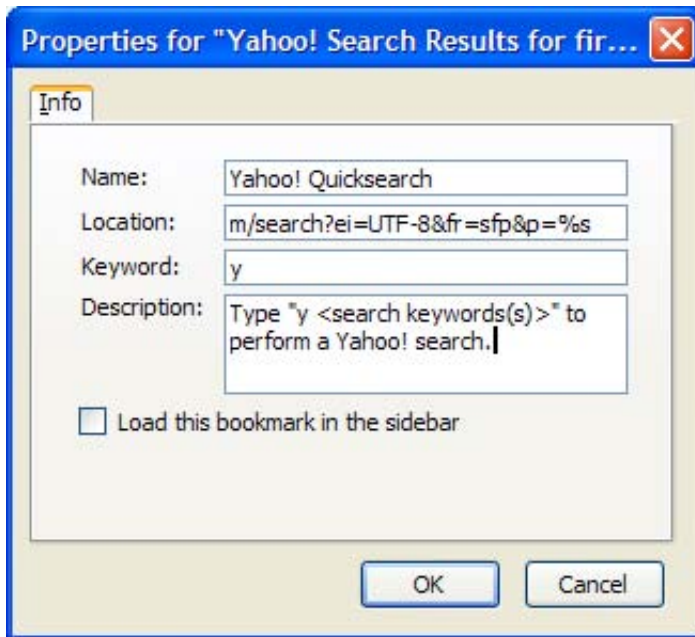
Finally, click OK, and there you have it: a Smart Keyword for Yahoo! Search. Try it out for kicks.

Now that you’re armed with enough knowledge to create Smart Keywords the hard way, you can use it to do some innovative and interesting stuff. Let’s create a currency conversion Smart Keyword that converts from European Euros to US Dollars. Using the service at <http://www.xe.com/ucc/>, let’s try to do this the hard way.

It turns out that this task is even more challenging than usual. Web developers will understand me when I say that the xe.com currency converter normally uses HTTP POST¹² for form submission, instead of HTTP GET, so there isn’t a URL that we can exploit: one into which we can insert the “%s” placeholder we used before. The Smart Keyword system can only use URLs based on GET requests. Non-developers probably won’t understand this technicality, nor should they have to. The upshot is that we need to find a GET request that we can use as the basis of our Smart Keyword.

¹²POST (or rather, HTTP POST) is a method of passing data to a Web page (or rather, a Web server).

Figure 2.21. Specifying the Yahoo! Quicksearch Smart Bookmark properties.



To do so, we have to dig into the HTML source of the Web page that performs the search (from the menu, choose View > Page Source, then start tearing your hair out!). With a little digging into the HTML source of the <http://www.xe.com> Web page, we can deduce that a useful Smart Keyword URL would be: <http://www.xe.com/ucc/convert.cgi?Amount=%s&From=EUR&To=USD>

We bookmark that URL, and edit the bookmark properties, remembering to give it a keyword: “e2u” is a good choice. To test out the new keyword—for example, to find out how much one million Euros is worth in USD—type **e2u 1000000** in the location bar.

If you know how to sift through HTML source, life is great; if you don’t, life isn’t so terrible. A little help from someone else is all you need. Here are some other Smart Keywords that you might consider.

Google I’m Feeling Lucky

<http://www.google.com/search?q=%s&btnI=I'm+Feeling+Lucky>

AltaVista

<http://www.altavista.com/sites/search/web?q=%s>

Amazon

<http://www.amazon.com/exec/obidos/external-search/?index=blended&keyword=%s>

Google News

<http://news.google.com/news?q=%s>

Google Groups

<http://google.com/groups?q=%s>

eBay

<http://search.ebay.com/%s>

Feedster recent blog posts

<http://feedster.com/search.php?q=%s&sort=date>

The Internet Movie Database

<http://imdb.com/find?q=%s>

Thesaurus.com

<http://thesaurus.reference.com/search?q=%s>

CiteSeer (Computer and information science publications database)

<http://citeseer.ist.psu.edu/cis?q=%s>

The basic premise behind the Smart Keyword feature is very simple: it simply replaces the “%s” placeholder with the keyword that you type in. Armed with this knowledge, you can make a Smart Keyword for just about any Web service that has a variable component in its URL.

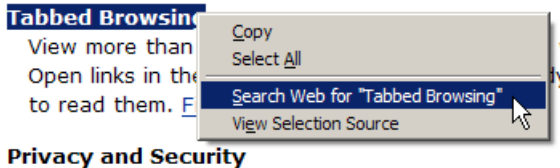
Right-Click Text Searching

I started this section by saying that TIMTOWTFI (There Is More Than One Way To Find It). Then, we searched for the text “blogosphere.” That technique is called “text search,” and it’s something that bears a little further exploration.

To perform a text search, you must master two mouse gestures: highlight, and context select. Highlighting text is easy—for simple operations, at least. To do so, click before the required piece of text, then, holding the left mouse button down, drag the mouse over the text to create the highlight. Release the mouse button to finish the highlight. When you do the context-selection (right-clicking

on the highlighted content), you'll see a Search Web for "*whatever you selected*" item appear in the context menu, as shown in Figure 2.22.

Figure 2.22. Convenient search from the context menu.



If you left-click on this menu item (I dare you to), Firefox will perform a search for the selected text on Google in a new background tab. Yes: on Google, again!

Text search isn't restricted to text, either. If you highlight an image on the page (by clicking *just outside the left edge of the image*, then dragging across it), the context menu will provide a search based on any `alt` (descriptive) text that's specified with the image. Alas, if the image is provided as part of a style, you can't do this yet; the image must be a proper part of the page.

You can also perform a highlighted search for text that appears in a link. Highlighting text in a link is quite tricky, though: you might accidentally start a download or open a tab if you try to drag-highlight across such text. Instead, start by highlighting the text with the Find Links As You Type feature (discussed next as part of the section called "FastFind: Find As You Type"). Once the text is highlighted, bring up the context menu by right-clicking on it as you would for any highlighted text.

As yet, you can't perform a text search on text inside a textbox or a text input field, but that feature may come in future.

What if you don't want to use Google as the default search engine? A bit of geek-speak is required to change that. You might like to read Chapter 6 before experimenting, though.

Type **about:config** into your Firefox address bar, hit **Enter** and look for the `browser.search.defaulturl` preference in the displayed set of preferences. Its value should be set to: `http://www.google.com/search?lr=&ie=UTF-8&oe=UTF-8&q=.` Replace this string by right-clicking on the preference. Use `http://search.yahoo.com/search?p=` for a Yahoo! search, or the equivalent string for another engine. Note that "%s" is not required in this case.

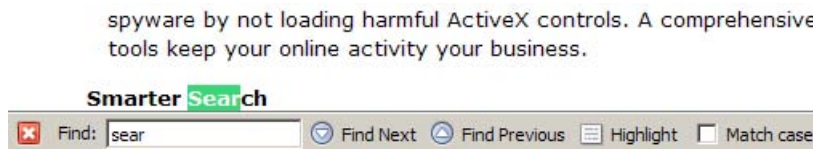
FastFind: Find As You Type

Other than searching for a Web page, you will often find yourself searching *within* a Web page. This is especially so for Web pages that contain lots of (largely irrelevant) content, for long listings, and when you're looking for a specific piece of information.

Firefox has an easy to use über-feature called **FastFind**. This is the official marketing buzzword; it's also known among veteran Firefox users as **Find As You Type** (FAYT), or **Type-Ahead Find**. Find As You Type does exactly what it says: it finds text *as you type*. How does it work? Just hit / (forward-slash) or **Ctrl-F**, then start typing the word (or words) you're looking for. Firefox will find and highlight the first instance of the word(s) that matches what you've typed, from the second you start typing. The easiest way to see what I mean is to try this out for yourself. Go to the Mozilla site²⁰ and type **/firefox** as soon as the page has finished loading.

You can also use **F3** or **Ctrl-G** to find the next match on the page, and **Ctrl-Shift-G** to find the previous match. So there's no need to take your hands off the keyboard, and no need to deal with a pesky Find dialog window that gets in the way! Instead, you get an unobtrusive Find toolbar at the bottom of the page, as shown in Figure 2.23.

Figure 2.23. The Find toolbar.



This is one of my favorite Firefox features. When I occasionally use Internet Explorer, I often lapse into hitting / and typing text in an attempt to search for stuff. Alas, this feature doesn't work in IE...

If you're daunted by having to remember keyboard shortcuts, have no fear: the Find toolbar is here (so much for poetry). Whenever you start a find, the toolbar springs into existence at the bottom of the page. You can use the buttons on the Find toolbar to find the next or previous matches. I recommend you get used to the keyboard shortcuts though, because they really speed up the searching process.

²⁰ <http://www.mozilla.org/>

Perhaps you noticed the Highlight button in the Find toolbar. You can use this to highlight all text that matches your search word(s). The keyboard shortcut for this is **Ctrl-Enter**. Figure 2.24 shows what this looks like.

Figure 2.24. Highlighting “Firefox” through the Find toolbar.



This kind of highlighting uses yellow to mimic the action of a felt pen highlighter, and doesn't change your current selection. There's also a checkbox that you can check should you want to match case (i.e. typing **FiReFoX** will match "FiReFoX" but not "Firefox" or "firefox").

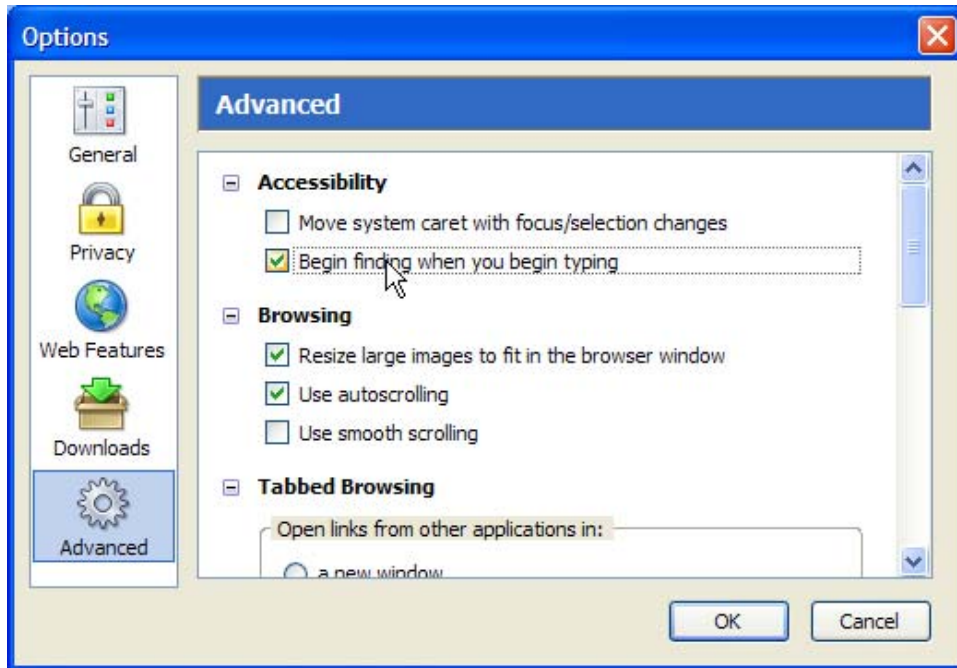


There are probably times when you want to search only within the text of links, ignoring all the other text on the page. A good example is my endless scanning for the "download" link on a Web page. To start searching only within links, hit ' (single apostrophe), instead of the usual / (forward-slash) or **Ctrl-F**, then start typing as you normally would. Only link text that matches your search term is highlighted. When you come to a link that you want, you can use **Enter** to load it in the current tab, or **CtrlEnter** to load it into a new tab.

A final variation that improves the efficiency of your searching is to enable the preference Begin finding when you begin typing, which can be found under Tools > Options > Advanced > Accessibility (Firefox > Preferences on Mac OS X, Edit > Preferences on Linux). That option's shown in Figure 2.25.

When this preference is enabled, you no longer have to hit / or **Ctrl-F** to start a search: just start typing your search phrase! This used to be the default behavior prior to Firefox 1.0, and is a well-loved feature among power users, yours truly included.

Figure 2.25. Begin finding when you begin typing.



Search is what you do with a browser when you want to find something. But, what do you do when you're presented with information that you'd rather not know? We take a look at this issue next.

Non-invasive Browsing

Firefox has built-in features that make your online browsing experience as smooth and uninterrupted as possible. Lots of features exist to remove the annoyances we tend to suffer when using IE. Popup and pop-under ads, spoofed status bar text that hides URLs, animated status bar text, flashing "Click me!" banner advertisements, windows that resize themselves to be either incredibly small or incredibly large: all these are annoyances. I have not seen any of them since I started using Firefox.

Annoyance elimination is the process of silently removing all these irritants. Annoyance elimination is one thing at which Firefox excels, so let's see how you

can configure Firefox to create the uninterrupted browsing experience that you deserve.

Popup Blocking

In the past, we lived with popup and pop-under windows that contained advertisements as if they were an unavoidable part of life. Fortunately, someone conceived of popup blocking software, which provided some relief. Eventually, software developers started building popup-blocking capabilities into their browsers; Firefox is one such browser. Popup blocking has become such an important feature of Web browsers that you'd now be hard-pressed to find one without it. Firefox, Safari, the Mozilla Application Suite, and even Internet Explorer on Windows XP (provided Service Pack 2 is installed) offer integrated popup-blocking features.

If you visit sites that use popups as a necessary element of the site's functionality (a banking site, for example), or you simply have some strange popup ad fetish, you can turn off Firefox's popup blocker, either globally, or on an individual site basis. We'll see how to tune the Firefox popup blocker to suit your requirements as we move through this discussion.



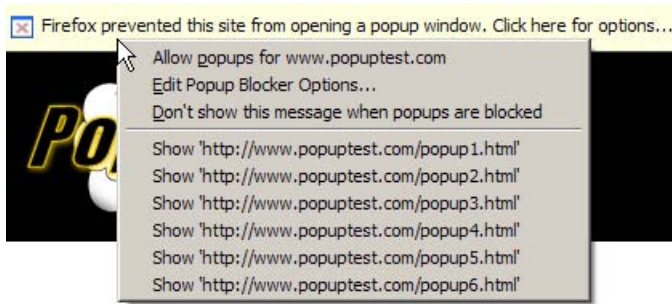
This is not the place to discuss the “ethics”¹³ of popup blocking and online advertising, so throughout this discussion I'll assume you'll want to block popups.

Avoiding Popups

Firefox is configured out-of-the-box to block popup windows, so you don't need to do anything to get this working. You can test out the popup blocker by going to a Website that uses popups. I suggest searching for “popup test” and visiting one of the popular testing sites. Whenever Firefox encounters a Website-initiated popup, as opposed to a user-initiated popup (more on that shortly), it will block that popup and display a notice that it has done so. You can see an example of the notice in Figure 2.26: the strip across the top appears by default; the displayed menu appears if you click on that strip.

¹³What has popup blocking got to do with ethics? Well, people argue that since online advertising is what funds many free (and non-free) Websites, popup blocking is like stealing from these sites, since they don't get revenue from your viewing of their content. To debate these issues is beyond the scope of this book.

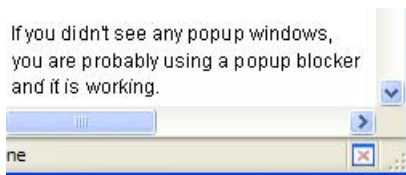
Figure 2.26. Firefox’s popup blocker generating a notice and blocked popup menu.



In this example, Firefox has dutifully blocked a popup from `http://www.popuptest.com/`. If you want, you can choose from the context menu the option to permanently allow popups from that site; additionally, you may (just once) choose to load the blocked popups.

When popups are blocked, you will also see an icon at the bottom right corner of your Firefox window: this will always be visible, even after you select *Don't show this message when popups are blocked*. In Figure 2.27 the icon appears to the left of the window resize grip (the small triangle of dots in the bottom-right corner).

Figure 2.27. The blocked popup icon.



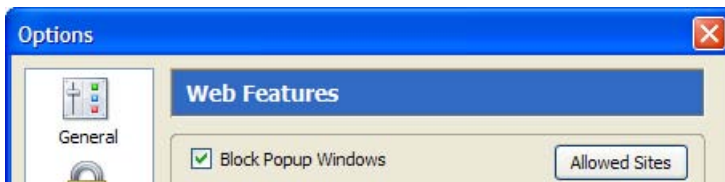
To view any blocked popup, click on the information bar and select the appropriate *Show 'http://www.example.com/popup.html'* option. The blocked popup will burst into existence for one time only. This is useful if you want to be sure you aren't missing anything (like a good deal on Viagra).

You can also tell Firefox to allow the Website to open popups from this point forward for a particular site. This is best used when you trust the Website and don't mind seeing its popup windows—perhaps it's a work-related Website whose

popups you should see. Whatever the reason, you can view popups for a given site by selecting Allow popups for *www.example.com* from the information bar.

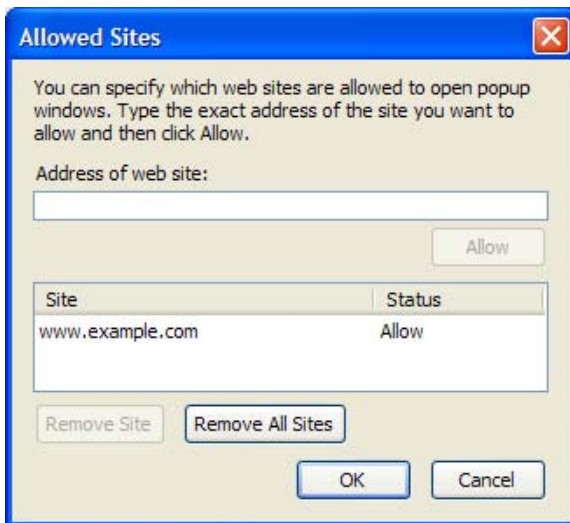
Don't be afraid to allow popups: you can just as easily disable popups again. To do so, access the Popup Blocker Options. Under Tools > Options > Web Features, you will see a Block Popup Windows checkbox, with an Allowed Sites button beside it. Figure 2.28 shows this feature.

Figure 2.28. Popup blocker options.



Click on the Allowed Sites button to bring up the list of sites that are allowed to open popup windows in your browser. An example setup is shown in Figure 2.29.

Figure 2.29. The Allowed Sites dialog.



This dialog holds a simple whitelist of sites whose popups are allowed to appear in your browser. You can add Websites to the list by typing their URLs into the Address of web site field in the dialog, but that's the hard way to do it. It's much

easier to allow popups from the information bar, as we did before. Removing a site is as simple as selecting it in the list (left-click), then clicking the Remove Site button.

User-Initiated Popups

A small note about popup blocking: Firefox doesn't block user-initiated popups. Such popups occur if you click on a link that spawns a popup. If you do this, the popup isn't blocked, because you asked for it. This is logical behavior: if I click on the link, I probably want it to do what it's supposed to do. See the Tabbed Browsing discussion in Chapter 3 for a way to trap even these popups inside an existing window or a tab.

User-initiated popups are often implemented using JavaScript. Next, we'll see how to control what JavaScript can and can't do.

Disabling Annoying JavaScript

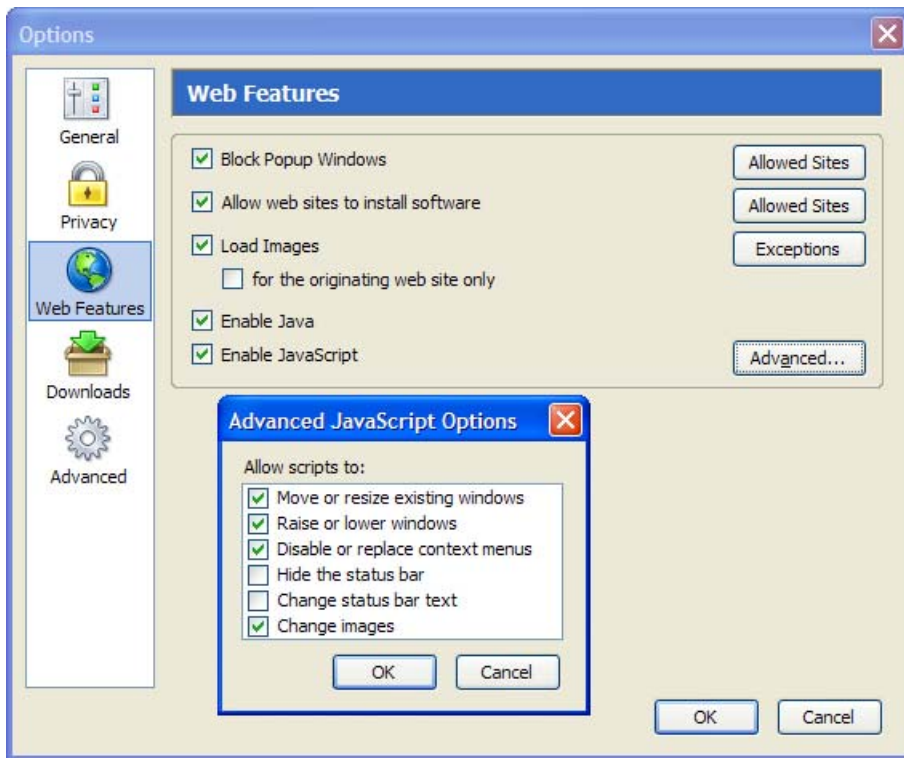
I don't know about you, but all those pesky things that Websites do with JavaScript really annoy me. My pet-hates are popup windows that spawn more popup windows, animated status bar tickers that obscure the normal status bar text, and spoofed status bar text that tries to mislead.

I'd always wished there was a way to turn off those particular JavaScript effects in my browser without losing the rest of the dynamic JavaScript functionality offered by the Websites in question. Preventing a Website from spoofing the status bar text is good. Preventing a dynamic form from using its JavaScript scripts to calculate whether I am "the life of the party" or the "uninvited nerd" is not so helpful.

Firefox supports this distinction nicely by blocking certain JavaScript calls while allowing the rest. Exactly which JavaScript calls are blocked can be configured in the Web Features preferences pane.

JavaScript Web Features

To find these JavaScript settings, look in the Web Features pane of the Options or Preferences dialog box. There is an Enable JavaScript checkbox alongside an Advanced... button on which you can click. Figure 2.30 shows this arrangement and the resulting secondary dialog box:

Figure 2.30. Advanced JavaScript options.

Don't be afraid to play with the options here just because they're described as "Advanced." These are actually fairly obscure options, rather than particularly advanced options. Here's a run-down of what each does.

Move or resize existing windows

Uncheck this to disable JavaScript that tries to resize a browser window that's already open. Sites that do this usually want to make the window smaller. They're often Web design sites, but can simply be sites that were created to annoy the heck out of you. This option will not prevent a popup window from choosing its own size, however. Popup windows can choose their own size provided that you've allowed popup windows, or the window is a user-initiated popup.

Raise or lower windows

This option is a little ambiguous at first inspection and, initially, I had some trouble figuring out what “raise” and “lower” actually meant! These options control whether JavaScript can be used to bring a window to the front (that’s “raise”) and, therefore to the top of the windows displayed on the desktop, or, conversely, to hide a window under others (that’s “lower”). A common example of the “lower” case, typically used for devious purposes, is the pop-under window, where a Website tries to open a secret window that you won’t notice. Sometimes this feature is needed, though, especially when Web-based applications need to maintain several open windows at once, or windows need to display warning messages.

Disable or replace context menus

Have you ever come across Websites that prevent you from right-clicking? No? Perhaps you remember trying to save an image that caught your fancy, only to find that you couldn’t right-click; instead, you’re told sternly that the image is copyrighted and not for download. When the Disable or replace context menus option is checked, that kind of thing may happen. Unchecking this option disables any JavaScript that tries to hide the context menu, so that the context menu is always free for you to use.

Hide the status bar

Uncheck this option to prevent Web pages from hiding the status bar at the bottom of the Firefox window. This is a highly recommended option since an absent status bar creates a security risk (see the next point).

Change status bar text

Some Websites spoof the status bar text, replacing the actual URLs of links with other URLs, or other text. Some sites have annoying animated stock ticker-like text that screams “WeLcOmE tO mY hOmEpAgE!” This is not only annoying but a security risk, because you can be fooled into clicking a link that goes to a URL that you weren’t expecting to visit. A carefully crafted piece of status bar text can also mislead you rather than annoy you. This is a so-called “phishing” scam that attempts to mimic normal browser functionality. I highly recommend you leave this option unchecked.

Change images

Unchecking this option disables features like JavaScript rollovers and dynamic menus that rely on images. It also disables some obscure form submission techniques based on image replacement.

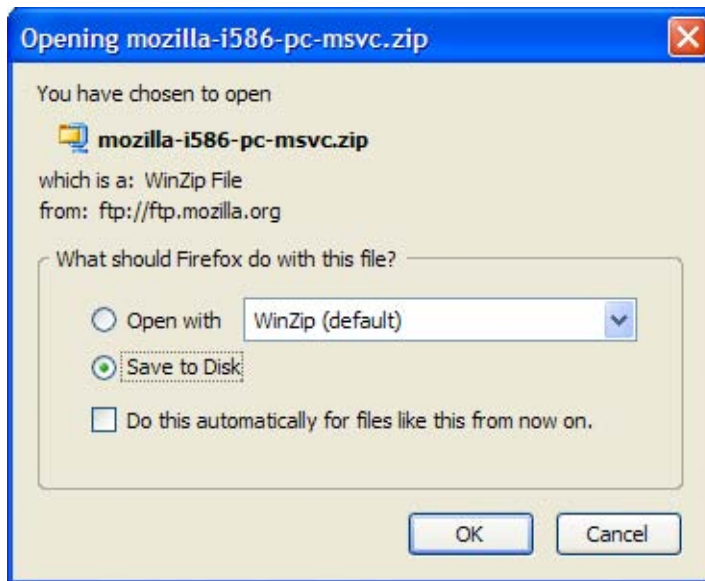
Downloading for Dummies

Using Firefox to download files from the Internet is a pretty common activity. It helps if the browser makes this mundane stuff hassle-free. Firefox has several standard features that make downloading easier; we'll step through these now. Later in the book we'll see how to enhance that download support with extra features.

Downloading Files

When you first download a file, Firefox displays a dialog window that asks you what it should do with that file. Choosing to open the file instructs Firefox to download the file to a temporary directory and then open it with the application you selected in the drop-down. As you'd expect, saving to disk simply saves the file to your hard disk. Figure 2.31 shows this initial dialog box:

Figure 2.31. The file download dialog.

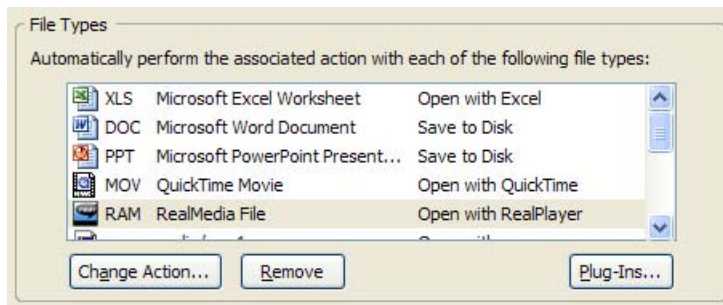


Downloads that are risky, such as executable files, cannot be opened by Firefox automatically: you can only save such files direct to disk. In such cases, the Open

with option is disabled, as you can see in Figure 2.31. This restriction prevents you from accidentally running an application that you did not intend to run.

Notice also the dialog checkbox labeled *Do this automatically for files like this from now on*. This option instructs Firefox to remember your preference and to pre-select it the next time you download a file of the same type. You can change this behavior in the Downloads section of the Options dialog box (Tools > Options, Firefox > Preferences or Edit > Preferences, depending on your platform). Figure 2.32 shows this dialog box after a few common file types have been configured.

Figure 2.32. Configuring file type associations.



Once Firefox has recorded your initial preference, you can change the default action taken for a remembered file type. Clicking the *Change Action...* button will allow you to change the application with which Firefox opens files of this type, or lets you tell the browser to revert to saving them directly to disk.

The Download Manager

Firefox comes with a Download Manager that displays all of your downloads in one place. Instead of displaying a download dialog for every single download, as Internet Explorer does, Firefox gathers your downloads together in a single location where you can track their progress without having to contend with multiple windows. Figure 2.33 shows the Download Manager.

Figure 2.33. The Download Manager in action.

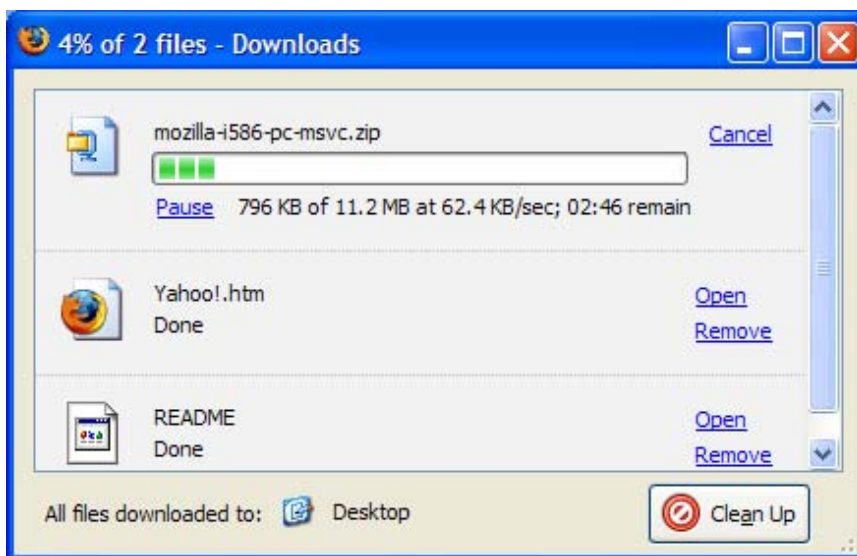
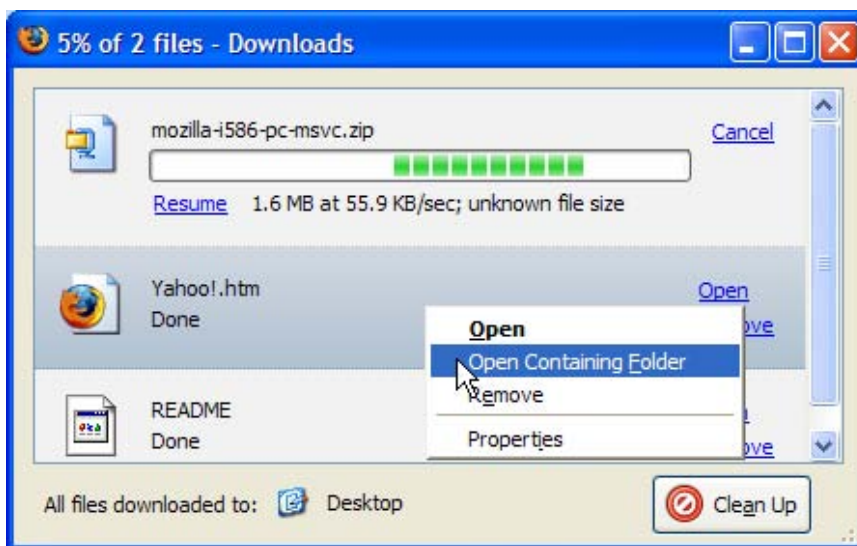


Figure 2.34. Opening a folder containing a downloaded file.



Let's quickly discuss exactly what you can do in the Download Manager. Firstly, you'll notice that downloads in progress can be paused or cancelled. Resuming paused downloads only works if the server supplying the file is configured that way, so don't pause if you're unsure of the server. Completed downloads can be opened by clicking the Open link, or double-clicking the row in which the file is listed. Completed downloads can also be removed from the Download Manager. Note that Remove doesn't delete any files from disk: it simply removes them from the Download Manager. Another commonly needed action is to open the folder containing the downloaded file. This can be done by right-clicking the row on which the file is listed in the Download Manager, and selecting Open Containing Folder. Figure 2.34 shows that detail.

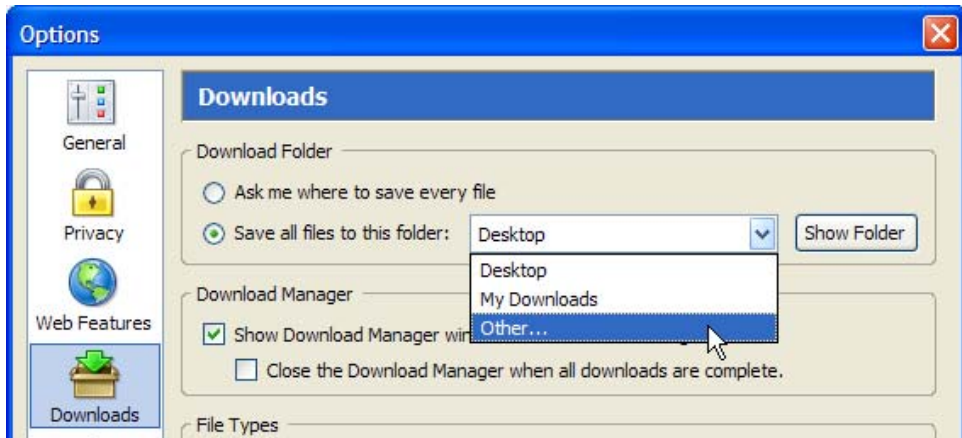
Notice that, at the base of the window, there's also an indication of the standard download directory. This is correct provided that you haven't configured Firefox to ask you where to save each downloaded file. In the screenshot above, the Download Manager helpfully reminds us that files are downloaded to the desktop, which is a folder or directory associated with the current operating system user account.

Finally, the Clean Up button is used to remove completed and cancelled entries from the Download Manager, which helps to keep the list of entries in the Download Manager to a sane limit. When all downloads are complete, Firefox pops up a small notification window in the bottom-right corner of the desktop, as shown in Figure 2.35.

Figure 2.35. The Downloads Complete notification.



Firefox saves all downloads to the desktop by default, and does so without prompting. If you prefer to save to another location, or to have Firefox prompt you for a save location for every new download, go to Tools > Options > Downloads (on Windows) and change the settings in the Download Folder section shown in Figure 2.36.

Figure 2.36. Configuring Download Folder options.

To be prompted for a save location each time you download, make sure the Ask me where to save every file option is selected. Otherwise, if you'd like to change the automatically selected default download location, simply select Other... from the drop-down, and choose the folder you want.

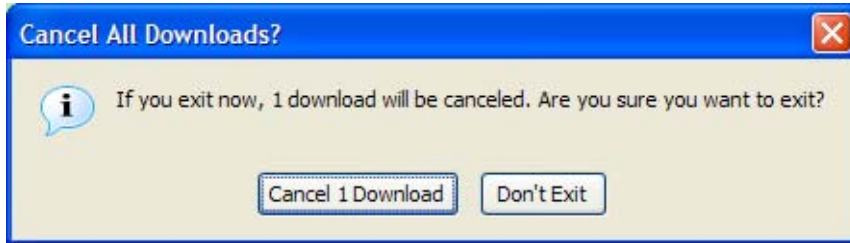
If you find the Download Manager a little annoying (in which case, you're not alone) you can disable it through the Downloads preferences (Tools > Options > Downloads on Windows) by unchecking Show Download Manager window when a download begins. If you do so, you can still call up the Download Manager from Tools > Downloads or via the **Ctrl-J** shortcut key. You will always be notified when all downloads are complete.

If you press Pause in the Download Manager while downloading a file, you can resume that download anytime as long as Firefox is still running. The Download Manager doesn't support the cross-session resumption of downloads as yet. **Cross-session resuming** is a rather useful feature that allows you to close Firefox, or even shutdown your computer, while downloading a file. When you start up again, the download picks up where it left off. This feature is targeted to appear in Firefox 2.0.¹⁴ While we (especially those of us on dial-up) wait patiently for this killer feature, you may be comforted to know that at least Firefox protects your in-progress download with a warning if you try to exit the browser prematurely. The dialog in Figure 2.37 shows that warning, which provides you with

¹⁴See http://wiki.mozilla.org/index.php/Firefox:2.0_Download_Manager for more.

an opportunity to reverse your hasty exit decision. If you keep going, your partially downloaded files will have to be re-downloaded from scratch.

Figure 2.37. The partial downloads cancellation warning shown on exit.



Installing Plugins

Plugins are add-on programs that allow you to view non-HTML content such as PDF files, Flash content, Java applets, and video within Firefox browser windows, or in separate windows created from Firefox windows.

If a plugin is missing when you view a page that carries plugin content, then you'll see a jigsaw piece in place of the content; a yellow plugin information bar will also appear at the top of the page. Figure 2.38 shows this arrangement when the Macromedia Flash plugin is missing.

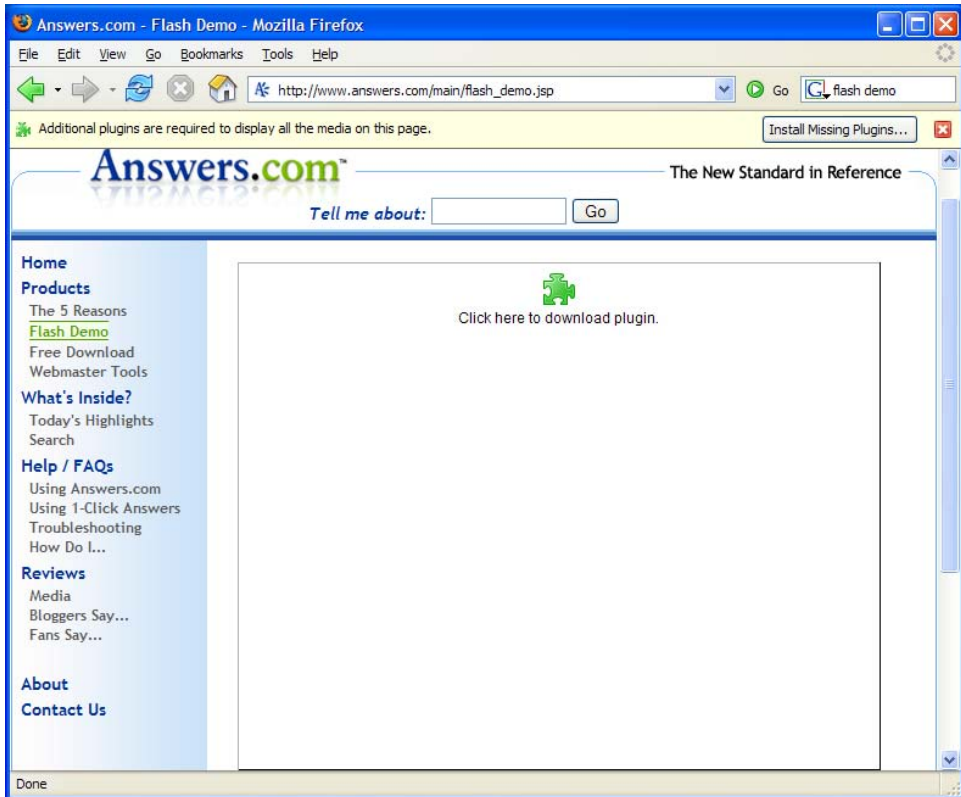
To install the required plugin automatically, just click the Install Missing Plugins... button and let Firefox find the plugin for you. Follow the resulting install prompts as you normally would. At the end of that process, the current page is redisplayed with the plugin content presented by the new plugin.

Major plugins are available for Adobe Acrobat, Macromedia Flash, Java, Apple Quicktime, Realplayer, and Windows Media Player. Firefox knows how to get all of these plugins. However, if, by some chance, the plugin isn't found by Firefox, or you want a particular version of the plugin or some other special arrangement, you can install plugins yourself. You can obtain most plugins at <https://addons.update.mozilla.org/plugins/>.

To install plugin software from that page, look for the link that suits your operating system, click it, and follow the instructions on the resulting page. Supporting

documentation is also found at the PluginDoc project.²³ You can refer to that more extensive site should you have any problems.

Figure 2.38. The plugin content placeholder and information bar.



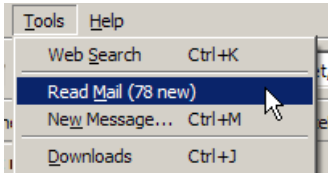
Getting to your Email

Firefox has an email integration user interface (UI) feature that lets you create and read emails from your browser. It's available on Microsoft Windows only. You can access this extra feature from the Tools menu: it even indicates to you how many unread emails you have, provided that Windows has registered a default email application. Of the two menu options supplied, choosing Read Mail will launch your default email application (probably Outlook, Outlook Express, or

²³ <http://plugindoc.mozdev.org/>

Thunderbird); choosing New Message... creates a new email message for you. Figure 2.39 shows these menu items.

Figure 2.39. Windows email options on the Tools menu.



If you choose to customize your toolbars (use View > Toolbars > Customize...), you can add an email button to the navigation bar. It contains a drop-down list that provides the same options for reading and creating emails. Figure 2.40 shows this button after it's been left-clicked.

Figure 2.40. Toolbar-based email button.



Summary

In this chapter, we've covered the core functionality of Firefox: the basic things you need to know to work efficiently with Firefox on the Web.

We took a good look at tabbed browsing and how it compares against the old-school non-tabbed browser interface. This author's conclusion is that tabs are a superior way to browse. Annoyance elimination is another task at which Firefox excels, and in addition to the standard features, you can also change popup blocking and disable bothersome JavaScript.

There Is More Than One Way To Find It (TIMTOWTFI) in Firefox, and by now you should be a master of search using Firefox. Smart Keywords are particularly clever, and so are you, now that you know how to create and use them! We also covered what I think is another killer feature of Firefox: FastFind, which allows you to search text within a page quickly.

On top of tabs, popups and search, we briefly covered working with non-Web content: downloading files, installing plugins, and accessing email. Firefox integrates quite well with all of those not-strictly-Web-browsing activities.

In the next chapter, we'll continue our exploration of Firefox's standard browsing features. We've already learned that Firefox is an efficient system for carrying you from the current page to the next page. In Chapter 3, we'll see that Firefox has many helpful features when the time comes to revisit a page that you've already visited.

6

Tips, Tricks, and Hacks

There are many subtle features hidden beneath the Firefox hood, many of which aren't accessible from the menu system: some even require that you learn hidden keyboard incantations! In this chapter, we dive into the deep end of the pool, where such wonders lie. We'll take a look at Firefox's hidden preferences, and uncover the hidden mysteries of your Firefox profiles. We'll also explore some useful tricks that do everything from speeding up the browser, to changing the way it looks and responds.

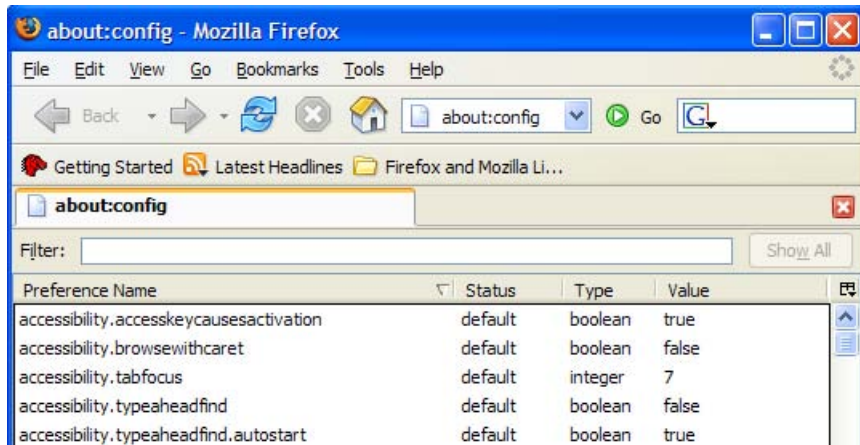
The Secret Named `about:config`

Firefox is designed to be immediately usable, without any configuration effort. There's no confusing glut of preferences and options in the user interface—a problem that plagues a number of other applications and Web browsers. By design, Firefox has fewer configuration interface intricacies than do comparable tools.

While this approach is vital for everyday use and for the “everyman” user, “power” users need something more flexible. Those special users are not left in the lurch, though, because Firefox has numerous extra preferences that one can set, albeit via a somewhat unofficial method. Such hidden preferences are managed through the **`about:config`** configuration page. This is a special interface to the Firefox preference system.

To call this interface up, type the special URL or Web address, **about:config** (yes, that's correctly stated) into the location bar, and hit **Enter**. You should see something like Figure 6.1.

Figure 6.1. Introducing the about:config interface.



Instead of an HTML Web page, the special preference system is displayed, using Firefox's own XUL page display language. The about:config page is a bit like a dialog box, except that it sits inside the existing window, rather than a window of its own. In Figure 6.1, the page is displayed inside a single tab.

Some explanation of this interface is in order. Each line in the list reflects a single preference. You can see five such lines in Figure 6.1. Each preference has a name and a value. The Status column of a preference indicates whether it is set to the Firefox default value (default), or has been changed by the user to something else (user set). The Type column of a preference identifies the kind of data that preference holds.

note

Preferences can be one of three types: string, boolean, or integer. String preferences are simply text strings (like "hello"), which include URLs (<https://update.mozilla.org/extensions/?application=%APPID%>). Boolean preferences are yes/no options; they hold either true or false. George Boole, a famous scientist, was the first to explain how to work with true/false values, so they're also called "Booleans." Integer preferences take on any integer value (such as 1, 2, 23, 1099), but can't accept decimal numbers like 5.4.

At the top of the page you'll see a Filter text field into which you can enter some text to filter out preferences. This is very handy, since the full list of preferences spans a good number of pages.

Modifying Preferences

Changing an existing preference is very simple. Just double-click on that preference's line, or right-click on it and select *Modify*. The values of Boolean preferences will be automatically toggled (swapped from "true" to "false" or vice versa). Modifying string and integer preferences causes a dialog box to appear, prompting you for the new value for the preference.

As you make your changes, you'll notice the status of the preference changing from default to user set, and that the preference text is displayed in bold type. This provides a clear indication of preferences that have been changed from the built-in browser defaults.

Among the hundreds of Firefox preferences there are some particularly handy ones that help to make Firefox even better than it already is. Let's discover a few of these hidden gems.

Speeding up Firefox

Here's a configuration tip that enables Firefox to download Web pages faster: enable HTTP pipelining. **HTTP pipelining** allows Firefox to send multiple requests at the same time, resulting in faster loading times for Web content.¹



HTTP pipelining is an experimental feature and can cause some Web pages to display incorrectly if they're not properly supported by the Web servers. While the speed improvements may be worthwhile, if your favorite Web pages start to look weird, try turning pipelining off again.

To enable pipelining, open the `about:config` interface and:

- ☐ Set the `network.http.pipelining` preference to “true” to enable HTTP pipelining for Web access that’s “direct connect.”
- ☐ Set the `network.http.proxy.pipelining` preference to “true” to enable HTTP pipelining when the Web is accessed via a proxy server. You only need to do

¹More technical details on HTTP pipelining can be found at <http://www.mozilla.org/projects/netlib/http/pipelining-faq.html>.

this if you're using a proxy server; your ISP or network administrator can tell you if that's the case or not.

- ❑ Set the `network.http.pipelining.maxrequests` preference to a number between **1** and **8**. This number is the maximum number of requests that Firefox may send out at any one time in a pipeline. I recommend it be set to the maximum of **8**² initially, then test working backwards from there. Step the value down by one (first to **7**, then to **6**, and so on) each time you notice the page display having problems.

With this hidden preference, you have to restart Firefox entirely before you can determine if a speed boost has been gained.

Inline URL Auto-completion

Here's a useful hack to enable inline auto-complete for URLs. When this feature is working, whatever you type into the location bar is automatically completed by Firefox with the closest possible match that can be determined from the browser's history. You can hit **Enter** when you get a match, or keep typing. This is much like the Inline AutoComplete feature in Internet Explorer. Figure 6.6 shows it at work.

Figure 6.6. Inline auto-completion in the location bar.



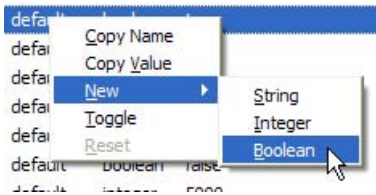
To enable this effect, you must add a preference that doesn't yet appear in the `about:config` list. Go to `about:config`, right-click anywhere on the page, and select **New > Boolean** from the context menu, as shown in Figure 6.7.

This menu choice yields a couple of dialog boxes. Enter **`browser.urlbar.auto-fill`** when prompted by the first dialog box, as depicted in Figure 6.8.

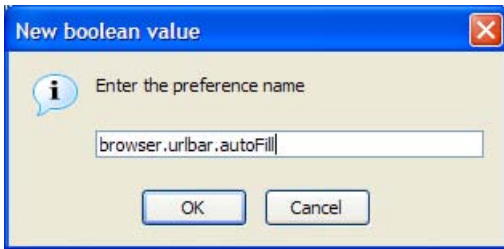
Select **true** when you're prompted to select a value, and that's it: inline auto-complete for URLs is activated! There's no need to restart Firefox in this case; it works straight away.

²The maximum is 8 as specified by the HTTP 1.1 specification; don't try to set it any higher, as this will have no effect.

default



Downloaded from <http://ajphaphysocpharm.sagepub.com/>



7

Web Development Nirvana



This chapter is geared towards Web developers and designers, and assumes some knowledge of Website creation.

From its penchant for Web standards compliance, to its built-in Web developer-friendly features, and onwards across the amazingly useful plethora of Web development-oriented extensions, Firefox is, without doubt, a far more useful Web development tool than any other Web browser! Such hyperbole, is, in this case, entirely accurate.

Firefox's checklist of useful tools is extensive. Friendly and actually useful JavaScript debugging tools—check. Provision of detailed Web page information—check. Real-time DOM inspector and editor—check. Ability to view HTTP headers as they are returned from a Web server—check. Validation of HTML, CSS, and accessibility conformance with minimal clicks—check.

Using extensions, you can even edit the CSS of a living, breathing Web page and observe the results immediately. It doesn't even matter which Website the Web page comes from!

In this chapter, we'll explore Firefox's Web developer-friendly features. We'll also delve into the extensions that make indispensable tools for so many Web

developers.¹ Read on if you want your development work to proceed as smoothly and professionally as possible.

Firefox's Standard Tools

You don't need to buy or download anything to get started. Many useful tools lurk inside the standard Firefox install.

Viewing Source Fundamentals

One of the first tasks with which many Web developers find themselves involved is examining the source of other people's Web pages, whether for debugging purposes, as a learning aid, or simply to satisfy their own irrepressible curiosity. Firefox comes with a nifty page source viewer that offers built-in syntax highlighting. Highlighting makes it easier to read the HTML source, and simpler to grasp the implied structure of the page.

You can view the source of a page using the menu system: choose View > Page Source. Alternatively, right-click and choose View Page Source from the context menu, or hit **Ctrl-U**. Figure 7.1 shows the scrutiny of a typical page's source; in this case, it's the Firefox homepage on the Mozilla Website.²

If you read and write HTML for a living, you probably already use an HTML editor that offers syntax highlighting. Modern browsers should perform to the same level. Yet Firefox's support is far superior to simply opening a black-and-white Notepad session—the default action for Internet Explorer 6.0.

Viewing Selection Source

Sometimes, viewing a whole Web page is just too much. Have you ever wished that you could check the HTML source of specific page elements on their own?² Perhaps you saw a well-designed element on someone else's Web page, and wanted to know how it was done. Using Internet Explorer, you would probably view the HTML source of the entire page, then use the find/search function to locate the

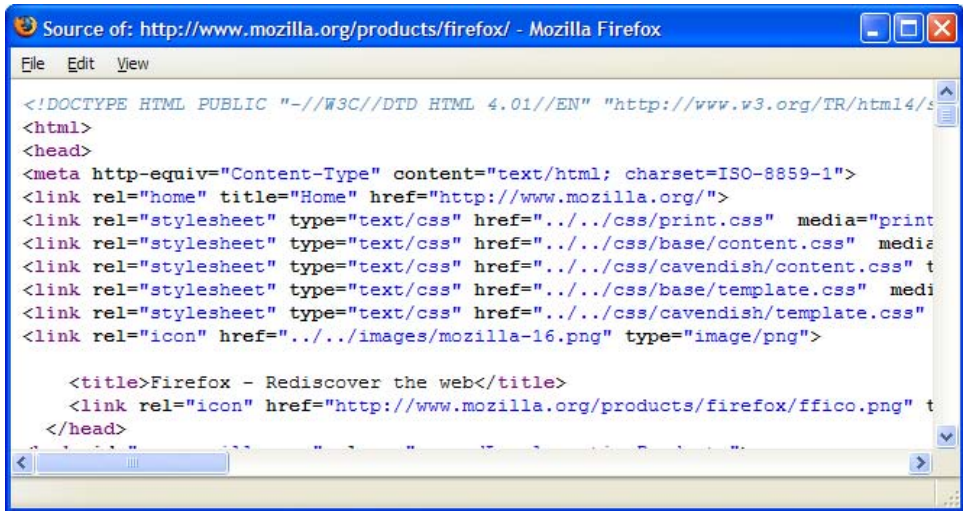
¹Including myself: I've been a freelance Web developer since 2001 and I'm currently employed as a Java Web developer.

²<http://www.mozilla.org/products/firefox/>

²Generating dynamic content is a common Web development task that's always prone to careless mistakes, or so I've found.

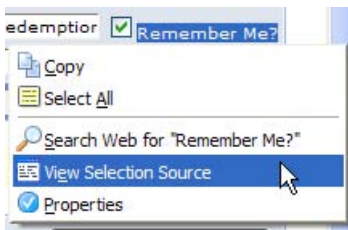
matching piece of HTML. This can be an unreliable process that depends on your recollection of contextual clues from the page, or other obscure hints.

Figure 7.1. Viewing the source of the Firefox homepage.



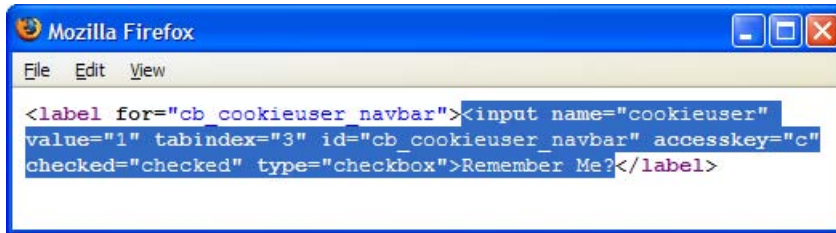
In Firefox, this process is hassle-free. Just use your mouse to highlight the interesting elements of the page, right-click, and then select View Selection Source. For example, if you want to find out how an input form's checkbox is coded, just use your mouse to select the whole checkbox; then, click the right mouse button and choose View Selection Source. That's shown in Figure 7.2, where the "Remember Me?" checkbox is highlighted.

Figure 7.2. Choosing View Selection Source from the context menu.



In return, Firefox will show you the HTML of your selection, and its surrounding code, as shown in Figure 7.3.

Figure 7.3. Displaying source code for the selected source.



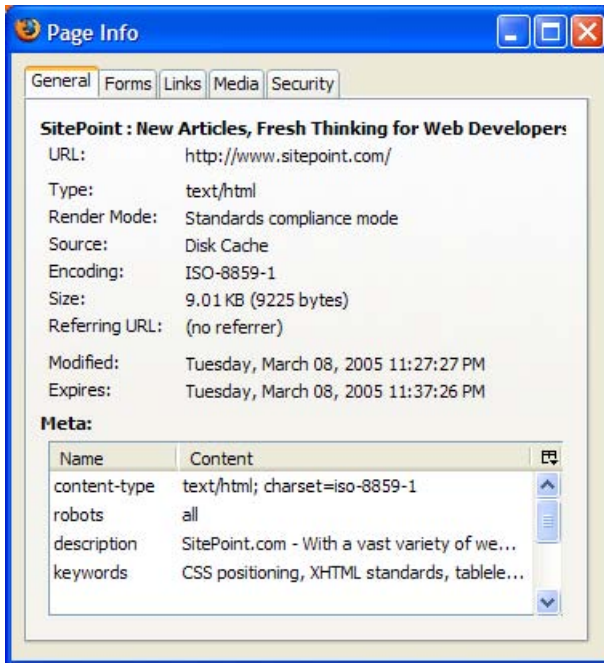
View Selection Source has saved me countless times as I've coded HTML forms, in which it's easy to make trivial syntax mistakes. View Selection Source lets me quickly check any form elements that don't seem to be working properly.³

View Selection Source is also a great tool for helping developers learn from the best. Often, I find myself looking at the source code of Websites created by the best Web developers and designers. To learn something new, I just view the source of the interesting part of the page in question. By narrowing down the HTML source to a specific area, View Selection Source saves time and keeps me focused.

Page Info Document Reports

Firefox provides very detailed information about Web pages via the View Page Info feature, which you can access from Tools > Page Info, or by right-clicking on the page and selecting View Page Info from the context menu. Either method pops up the Page Info window, shown in Figure 7.4, which tells you everything you'd ever want to know about the page.

³Often, the cause of the error is an incorrectly named form field, or the omission of a closing quotation mark.

Figure 7.4. Displaying Page Info for sitepoint.com.

Each of the tabs at the top of the window represents a logical area of information about the Web page. Let's go through each tab and identify which bits are useful to Web developers.

The General Tab

As its name suggests, the General tab shown in Figure 7.4 displays general information about the Web page, including its MIME type, character encoding, and the size of the page. The character encoding information is particularly useful for those working on internationalized Web pages. If you've had any experience developing Web pages in multiple languages (and, therefore, using multiple character encodings), you'll know that if you're not careful, you'll create incorrectly encoded pages that display as gibberish. Ensuring that you're using correct character encoding can save a lot of debugging time down the track.⁴

⁴For instance, just because you set the meta content-type to "iso-8859-1" doesn't necessarily mean the Web page is using that encoding!

Notice also the Render Mode field. This indicates how the Firefox layout engine (called Gecko) renders the Web page. In **Standards compliance mode**, Firefox renders the content exactly in the manner described in the HTML and CSS specifications. **Quirks mode**, on the other hand, accommodates existing and poorly made Web pages that are coded to older standards, or are not coded to standards. Pages displayed in Quirks Mode still look reasonable because their content is laid out and rendered in a way that aims to represent a “best effort,” rather than being “pedantically correct.” Firefox determines which mode to use by checking first the HTTP Content-Type, then the DOCTYPE of the page. Pages with Content-Type `application/xhtml+xml` or `text/xml` are rendered in standards compliance mode. Unlike Internet Explorer, Firefox always follows the HTTP Content-Type’s instructions.

Finally, the General tab provides a list of the `<meta>` content tags used in the Web page, presenting this information in a more readable format than is available through the HTML source.

The Forms Tab

The Forms tab is an interesting case: it shows the details of forms and form fields in the Web page. Figure 7.5 shows the Forms information displayed for Google’s home page.

The top listbox identifies the forms in the page; the bottom listbox shows the form fields contained in the form selected in the top listbox.



If the page doesn’t contain any forms, this tab will be empty.

This display is extremely convenient in instances in which you need to verify the field names contained in a form, and the values of hidden form fields. It provides the information in a format that’s much more digestible than the alternative: HTML “tag soup.”

The Links Tab

The Links tab lists all the links contained in the Web page. The list includes the usual hyperlinks as well as links to style sheets, and any links that specify related meta information.⁵ Figure 7.6 shows a sample report.

⁵These are tags of the form: `<link rel="linkName"/>`.

Figure 7.5. Displaying the Forms tab for Google.com.

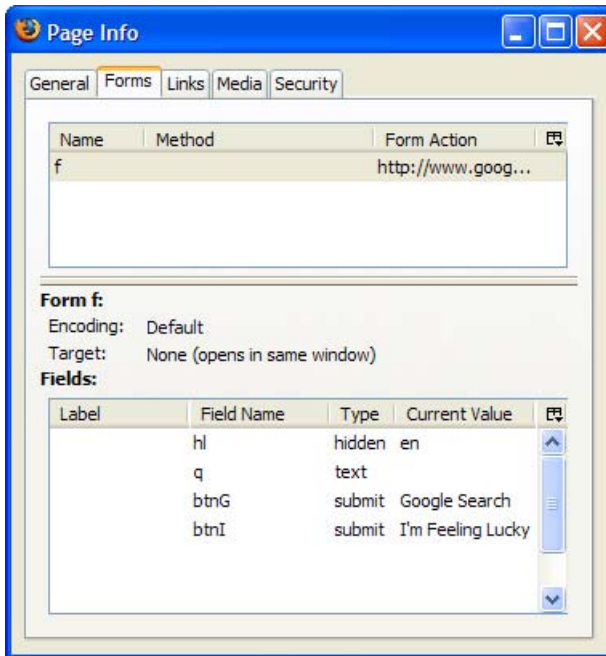
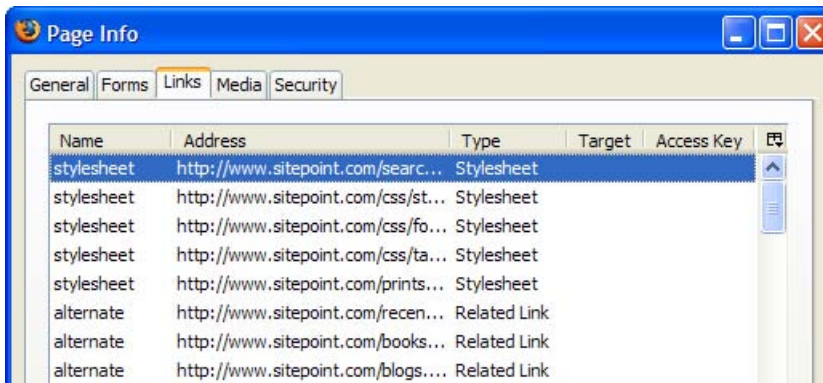


Figure 7.6. Displaying information about links on sitepoint.com.



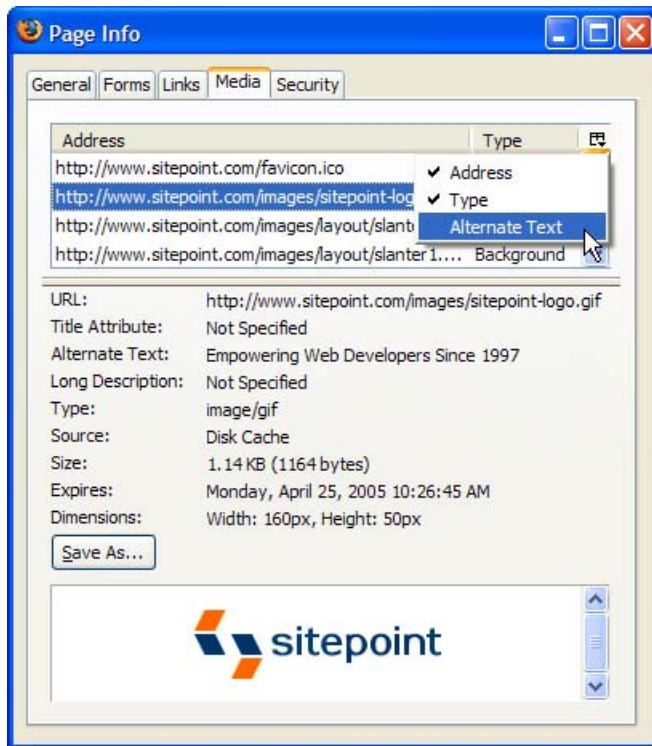
This display also lets you view each link's target, which identifies whether or not the link will open in a new page, and each link's access key, which is the keystroke that causes navigation for that link. You can display the Access Key column by

clicking the small button in the Page Info window's top-right corner, and checking the Access Key item in the dialog box that results. A list of access keys is very useful to those developing accessible Web pages, as it allows us to see, at a glance, if any links lack access keys, and if any access key is assigned to more than one link.

The Media Tab

The Media tab holds detailed information about all the media files (not just image files) embedded in a given Web page—even background images and favicons. You can view each media file's `title` attribute, as well as its alternate text. To enable the Alternate Text column, click the button in the window's top-right corner, and check Alternate Text, as shown in Figure 7.7.

Figure 7.7. Reviewing information about media files on sitepoint.com.



Displaying the alternate text for media files makes it easy to identify images for which the `alt` attribute is not specified, and remedy the situation. This is especially handy if, for some reason, you can't use online validators such as Bobby.⁷ It's also a good way to check that a page of good quality hasn't been damaged by small changes you've just made.

Another useful feature of the Media tab is that it gives you the ability to view and save media files. Simply select a given media file from the list, and a nice little preview of it appears, along with a Save As... button that lets you save the file. Sometimes, this is the only way to access an embedded media file, such as a Flash applet, that you want to save.

The Security Tab

The Security tab provides security-related information about a Web page. On a normally delivered Web page—one that's not sent via HTTPS (secure HTTP)—the Security tab will tell you that the identity of the Web site cannot be verified, and that it does not support encryption. On securely delivered pages, however, the Security tab shows you whether the Website is verified (by its certificate), and identifies the type of encryption the page uses to protect any transferred data. Figure 7.8 shows that information for Google.com.

Figure 7.8. Accessing encryption details via the Security tab.



⁷ <http://www.cast.org/bobby/>

This is the only way to access the encryption details for secure HTTP. For peace of mind, it's worth checking these details if your page—or your site—needs to integrate with some other site, such as a payment processing facility.

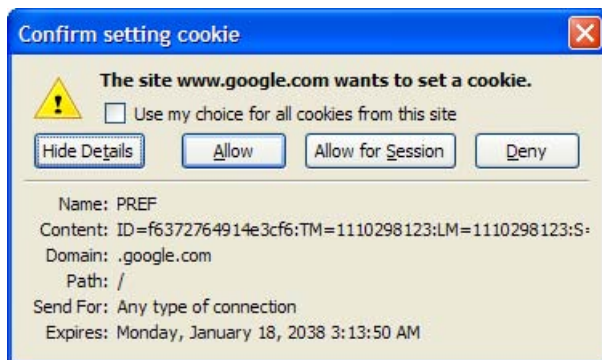
Popup Cookie Tracking

You may find it necessary to set a cookie in the user's browser. Naturally, you test the cookie by surfing through the matching Web pages and verifying that they're working as intended. Perhaps you print debug messages when cookies are set, so that you can see what's going on. Wouldn't it be nice if you could confirm whether cookies are set without having to add temporary diagnostic code? Firefox allows you to do just that.

To get this functionality working, go to Tools > Options > Privacy, and expand the resulting tab's Cookies section (if it isn't already expanded). Select Ask me every time from the Keep Cookies: dropdown, and click OK.

Now, surf to a Web page that sets a cookie: any Web-based email service should do the trick, as will the Internet Movie Database.⁸ Firefox will inform you that a cookie is being set, and if you click Show Details, you'll be able to access specific information about the cookie, from its name and value, to its expiry date. This information is absolutely invaluable for those debugging cookie- and session-based Websites. Figure 7.9 shows the details that Firefox reports for Google.com.

Figure 7.9. Firefox reporting that Google.com wants to set a cookie.



⁸ <http://www.imdb.com/>

Stretch the window to see the whole cookie value, if it's too long to fit in the default display.

Once the cookie is accepted—click Allow to make that happen—you can easily recall it later. To view the set of currently held cookies, use the options buried behind Firefox's menus, as discussed in Chapter 3. For cookies, access Tools > Options > Privacy > Cookies > View Cookies. On Linux, start with Edit > Preferences; on Mac OS X, start with Firefox > Preferences.

Viewing cookie details as they're set is something of a “divide and conquer” strategy for debugging Web pages. It allows you to handle problems one at a time, rather than have your cookie problems mixed up with the other issues on the page. You can worry about scripts that manipulate page content in response to cookie values after you've confirmed that the cookies themselves are in good order.

JavaScript Console Techniques

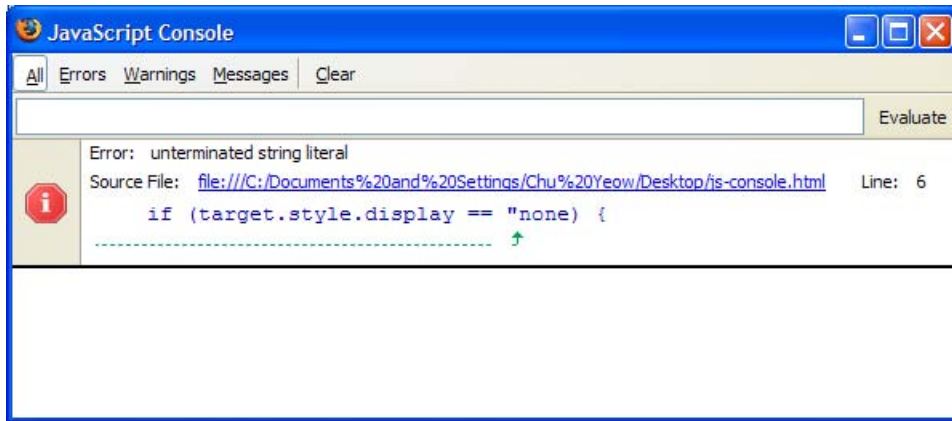
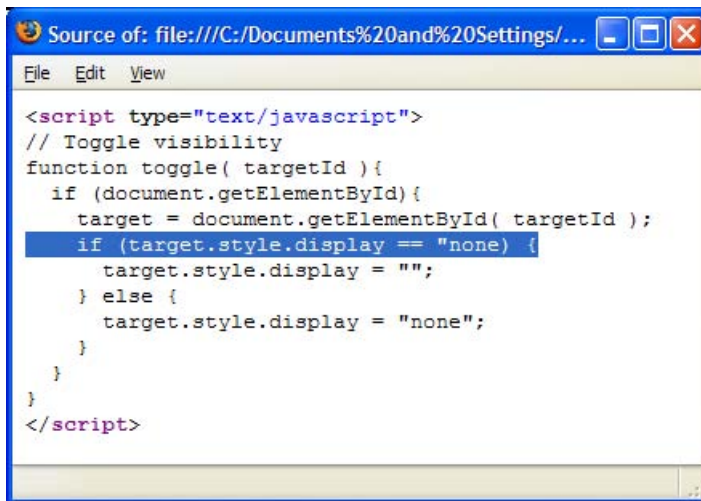
In Chapter 6, we briefly touched on the JavaScript Console. Let's now have a closer look at that tool. It's called a “console” because it's in this tool that any internally-generated messages from scripts will appear.

The console is a real gem—a lifesaver for those debugging JavaScript scripts. One brief attempt by the Firefox team to remove it from the browser for the sake of a smaller download met with such an outcry of protest that the plan was dropped immediately. The JavaScript Console is here to stay.

In helping debug your JavaScript, the console provides an error message, a direct link to the line of the JavaScript file on which the problem occurs, and, if applicable, the context of the error. Figure 7.10 shows the error generated when a string lacks its closing quotation mark.

The link that's displayed in the error message is an incredibly useful debugging aid. Clicking that link opens the View Source window, complete with a display of the problem page. Better still, the offending line is always centered—and highlighted—for your immediate consumption, as shown in Figure 7.11.

You can filter the entries shown in the JavaScript Console to show only errors, warnings, and messages. Click the text items that look like menu names to see how.

Figure 7.10. Displaying an error in the JavaScript Console.**Figure 7.11. Highlighting a syntax error in View Source.**

Two hidden preferences affect the JavaScript Console's operation. The first is `javascript.options.strict`, which is unset by default. Set it to true, and Firefox will sometimes report additional grumpy messages—usually around scripts that were written using poor coding practices. The second preference is `javascript.options.showInConsole`, which is set to false by default. Don't touch this unless you plan on becoming a Firefox extension hacker.

The JavaScript Console never pops up when an error occurs: you must always start it yourself, by hand. Press the Clear button before each page load to keep the list of messages to a bare minimum. If you do see the console pop up, this indicates that the current page is unwisely using the `javascript:` URL as a link target. That URL has no purpose other than to reveal the JavaScript Console. The problem page should use `href="#"` instead.

In Chapter 6, we touched briefly on the use of the JavaScript Console as a calculator. You can perform fancier scripting operations in the console if you work at it a bit. Try typing in the following simple script, then hit **Enter** or click the Evaluate button: **`var a=5; var b=6; if (a > b) alert("Impossible"); else a*b;`**.

The number 30 should appear. If you reverse the sense of this comparison, you should get an alert instead.

Finally, you can use this kind of “on the fly” JavaScript to probe the contents of a displayed Web page, even without the JavaScript Console. Let’s return briefly to the `javascript:` URL. Visit the Google homepage⁹ (or any form-based Web page), and type something into the search field (e.g. **Foo**), but don’t start the search. Instead, move to the location bar and type something along these lines: **`javascript:alert(document.forms[0].elements[1].value)`**.

In the case of Google’s home page, this input displays the search text that you just entered. You can look at any part of a displayed Web page in this way. For a more complete approach, try the JavaScript Debugger extension, which we’ll touch upon at the end of this chapter.

DOM Inspector Content Analysis

The DOM Inspector, or DOMi for short, is a tool that allows you to view and dynamically edit the Document Object Model (DOM) of XML documents such as Web documents or XUL pages. It’s an excellent tool for debugging and learning during Web development, and is almost essential for Mozilla application development (the process of writing Firefox extensions and themes).

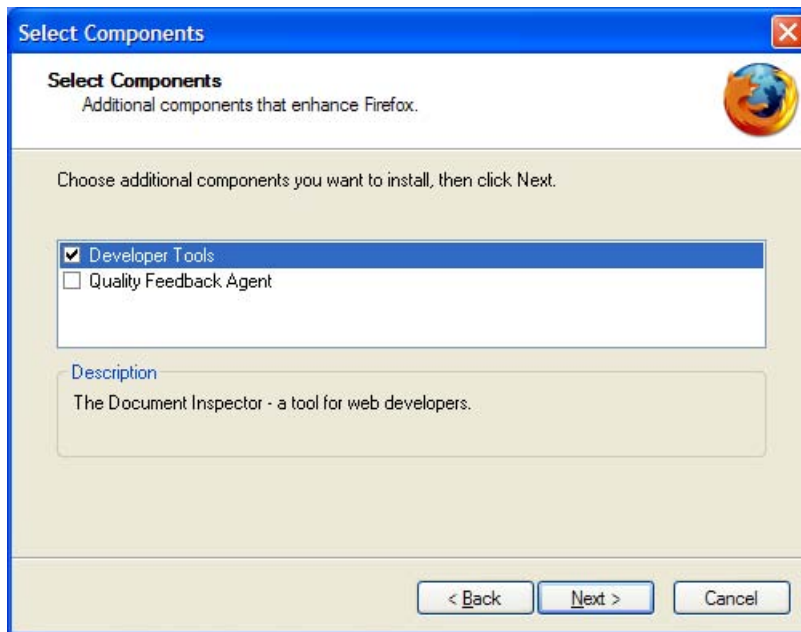
With the DOM Inspector installed, you can examine the DOM structure of an HTML document—the set of nested tags that make up the page. That structure is presented in a sideways-displayed, hierarchical tree format. You can zoom in on a particular element within that document, and study a broad range of inform-

⁹ <http://www.google.com/>

ation about it, including DOM Node object properties, the application of CSS style rules, Box Model properties, and even JavaScript methods and properties.

The DOM Inspector is not installed by default in the Windows version; to obtain it, you must select Custom Installation when installing Firefox. This provides the option to add the DOM Inspector, as shown in Figure 7.12.

Figure 7.12. Choosing to install the DOM Inspector.



If you forget to grab the DOM Inspector at install time, it's not the end of the world. A simple but somewhat disruptive solution is to uninstall Firefox, then re-install it, choosing Custom Install when prompted. A better solution is to download the Inspector as a separate extension.¹⁰ Make sure that you read the release notes carefully before you jump on the download link: at this time, Firefox version 0.9.85 is the absolute minimum version required.

Once you've installed the DOM Inspector, you can call it up via the Tools > DOM Inspector menu options, or via **Ctrl-Shift-I**.

¹⁰ <http://hacksrus.com/~ginda/venkman/>

Let's walk through the use of the DOM Inspector on the SitePoint homepage.

Opening a Document for Inspection

To begin inspecting a Web page, browse to that document in Firefox, then run the DOM Inspector (Tools > DOM Inspector or **Ctrl-Shift-I**). This will select the currently loaded document for inspection, and display the DOM of that document in the DOM Inspector. You can also select a document from the DOM Inspector's File menu by choosing a window (File > Inspect a Window), or by entering the document's URL into the DOM Inspector's text field and pressing Inspect, as shown in Figure 7.13.

Figure 7.13. Entering URLs for DOM inspection.



Let's load the SitePoint homepage.¹¹ The DOM Inspector will obediently populate the left-hand **Document pane** with a hierarchical tree view of the DOM of the page <http://www.sitepoint.com/index.html>. The right-hand **Object pane** displays information about the node selected in the left pane. The display is shown in Figure 7.14, after the left pane's content has been expanded a little by the user.

If the SitePoint page doesn't show at the bottom of your DOM Inspector window, add it via View > Browser. I find this a convenient way to match the node being inspected with its appearance in the document. If you prefer the DOM Inspector window to be less cluttered, you can leave the inspected page in a Firefox browser window. To do so, display the page in the normal browser window before opening the DOM Inspector.



There's only one version of the inspected page inside Firefox! The browser window and the DOM Inspector window display that single document in two different ways. This is called "multi-view display."

Selecting Nodes and Page Elements

Once the DOM Inspector is up and running, feel free to play with the page hierarchy shown in the top-left panel. Click on the little plus icons to expand and collapse any part of the hierarchy that seems interesting. This simple action reveals

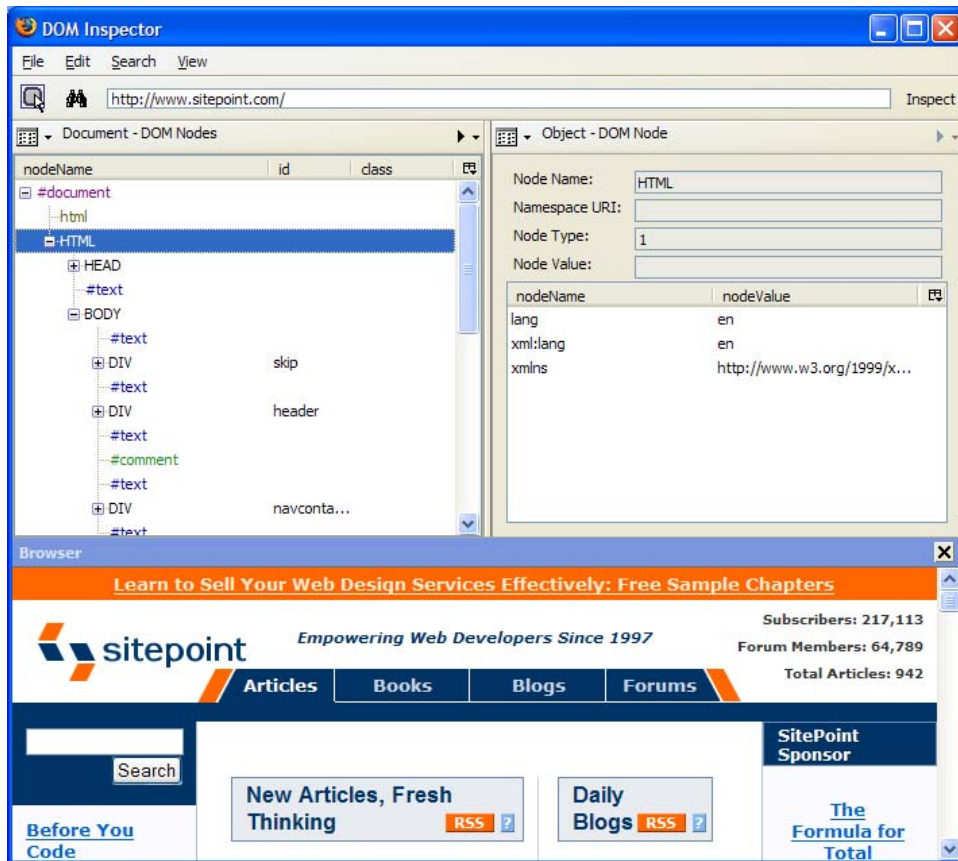
¹¹ <http://www.sitepoint.com/>

the structure of the page in a way that's separate, and different, from its visual layout.



If you aren't using the default Firefox theme, the plus icons might be replaced with some other icon. Regardless of what the icons look like, they operate consistently.

Figure 7.14. Inspecting the DOM of sitepoint.com.



Another way to access that same level of detail is to select a node for inspection using the Inspect node icon in the upper-left corner in the DOM Inspector toolbar. Selecting Search > Select Element By Click from the menu does the same thing. Figure 7.15 identifies that Inspect node icon.

Figure 7.15. The DOM Inspector's Inspect node icon.

Click the icon to activate the Inspect function, then bring to the front the browser window that holds the page. Click on the page element in which you're interested, and a flashing red rectangle should border the corresponding element (or its parent) within the page display. Return to the DOM Inspector window, where the page hierarchy will be expanded to reveal the matching node, which should be both selected and highlighted. This technique is particularly useful when you're trying to determine the properties of a given element of the document. The DOM Inspector lets you see everything without requiring you to add diagnostic styles or scripts to the page.

This element selection process also works in reverse. Selecting (by left-clicking) a node in the left pane of DOM Inspector will highlight the matching page node with the same blinking red border. That pairing effect is shown in Figure 7.16.

Figure 7.16. Highlighting a DOM Inspector element within the page.

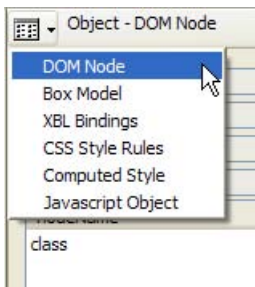
Finally, you can find specific elements by their ID, class, or tag. For example, you can search for <a> tags or tags, or even for the value of a specific attribute, like href="http://www.example.com/", using the binoculars icon, Search > Find

Nodes..., or **Ctrl-F**. Hitting **F3** will begin a search for the next matching element in the document.

Information Views

The DOM Inspector can display several types of information about a particular document element. These displays are available from the drop-down menu in the top-left corner of the right-hand panel. Figure 7.17 shows the items in that list.

Figure 7.17. Information types in the Object panel.



We'll take a look at each of these options in turn, and see how they can assist your Web or application development work.

The DOM Node View

The DOM Node view shows the DOM representation of the selected node. The view for the document container—the `<html>` tag—is shown in Figure 7.18.

We'll see later that this view is most useful when you're dynamically editing the DOM of the current document.

The Box Model View

The Box Model view shows positioning and layout-related attributes of the selected node. Values like the node's dimensions, padding, and margins are all shown. The small menu inside this view lets you pick which vital statistics are of most interest to you. Figure 7.19 shows the possibilities.

Figure 7.18. Displaying the document container in DOM Node view.

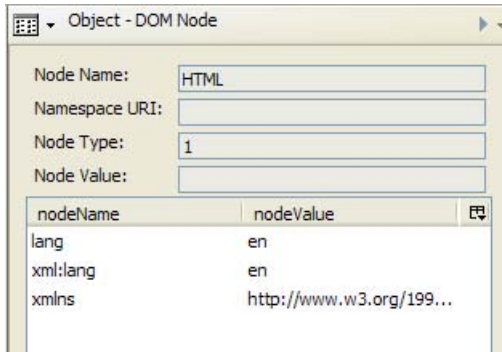
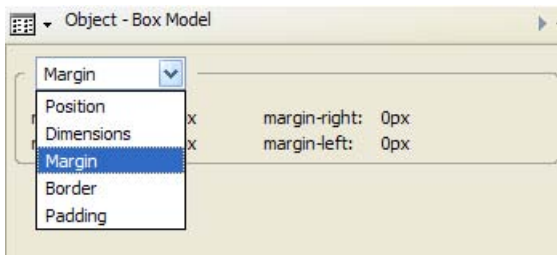


Figure 7.19. Inspecting positioning and layout attributes using the Box Model view.



This view is useful when you need to work out where any extra space comes from in a set of nested page elements.

The XBL Bindings View

XBL, or eXtensible Binding Language, is an XML markup language that attaches JavaScript objects and other content to HTML or XML elements in order to make those elements smarter and more functional. The code that's added is called a binding, and it describes the behavior of the bound element. XBL is slightly too advanced for this book, and is not yet a widespread technique within Web

development circles. Mozilla application developers—those that build extensions and tools that sit on top of Firefox—are the folks who are most interested in it.⁶

The XBL Bindings view shows the XBL bindings loaded into the inspected document by means of CSS rules. Use it when you're not sure what processing occurs when particular HTML events are targeted on the displayed page element, and you really want to know all the gory details.

The CSS Style Rules View

The CSS Style Rules view identifies the CSS rules that are applied to the selected node. It even shows you which fields the rules are defined in, and the exact line numbers at which they're located. This view lists the original rules—not the rules after they've been cascaded or computed. It's like View Source, but for CSS stylesheets rather than HTML.

You'll often see specified CSS rules that aren't included by the page itself. For example, there are usually CSS files of the form “resource://,” such as `resource://gre/res/html.css` and `resource://gre/res/ua.css`. These are Firefox's default CSS style rules, which are applied before any rules specified by the Web page. Figure 7.20 shows an example of such a rule in the first line of the top panel.

Select a rule in the top half of this panel, and its CSS properties will display in the bottom half; and—get this—you can add, edit, and delete properties, and observe the effects of your changes in *real-time* through this panel. In other words, you can observe immediately how your CSS rules will affect the layout, and look and feel of the Web page. Believe me, this is *extremely* useful for testing out experimental changes, or tweaking a page's layout. Since the changes occur in real time, there is no need to fall into the usual (and tedious) cycle of: edit file, save file, refresh page in browser, edit file again, and so on.⁷

Let's walk through a simple example that shows the convenience of this feature. On the sitepoint.com homepage, there's a little boxed element that acts as a header for the most recent SitePoint articles. It's shown in Figure 7.21 (but note that the SitePoint site is regularly reworked, so this may change).

⁶More information on XBL can be found at <http://www.mozilla.org/projects/xbl/xbl.html> and <http://xulplanet.com/tutorials/xultu/introxbl.html>, should you be interested in application development on the Mozilla platform.

⁷This would be even more useful if it weren't for the very convenient method of changing a page's CSS in real-time that we'll discuss later.

Figure 7.20. Identifying a CSS style rule in the CSS Style Rules view.

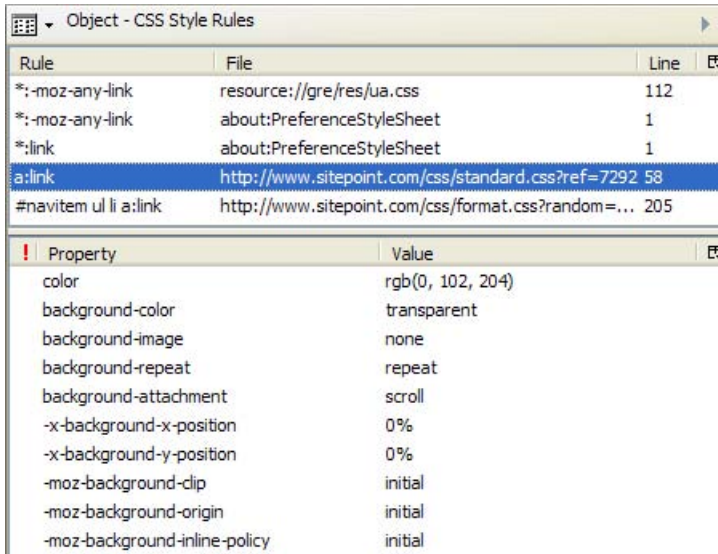


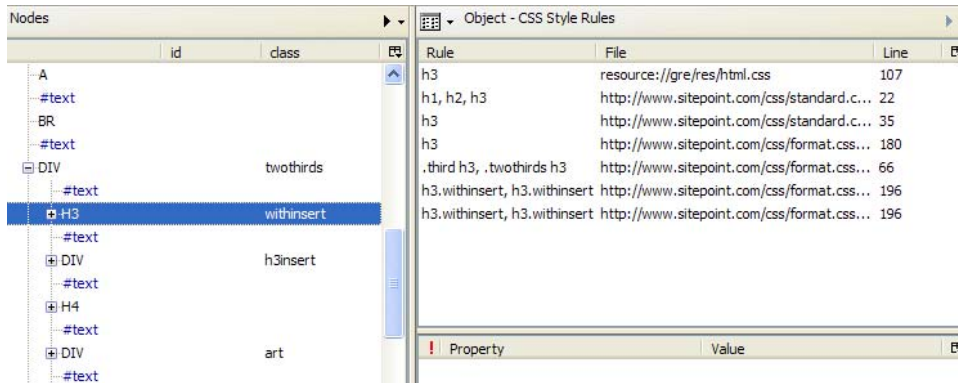
Figure 7.21. SitePoint's boxed title element.



Let's suppose that we're in the mood for tweaking, and want to experiment with center-aligned text in a serif font (Georgia looks quite good). Well, if we have direct access to the CSS files on the sitepoint.com Web page, we can make our changes there, upload the new CSS file, and refresh the page in our browser. Unfortunately, we don't have that access. This is where the DOM Inspector can be pretty nifty.

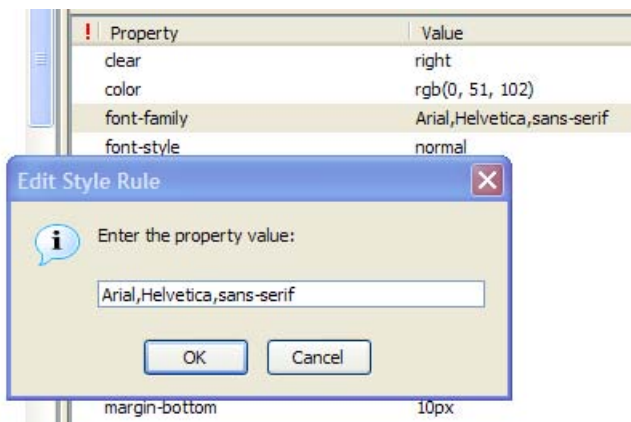
Using the DOM Inspector, we proceed in two steps: find the bit we want to change, then make the change itself.

First, inspect the page and select the element you want to edit. In this case, that's the boxed title element. Now, change to the CSS Style Rules view in the Inspector. You should see something like Figure 7.22.

Figure 7.22. Applying CSS style rules to the selected element.

The CSS Style Rules list shows the rules that apply to the selected node, in ascending order of specificity. In other words, as you move down the list, the rules become more specific, and therefore overrule the list items that come before them. Select the second h3 rule, and the DOM Inspector will fill the lower-right pane with the CSS properties applied from that file. Now you're all set up, and ready to make a change.

Let's change the font to our serif font of choice, Georgia. Find the font-family property in the list, and double-click it. You will see the prompt shown in Figure 7.23; this asks you to enter the property's value.

Figure 7.23. Editing a CSS property.

Enter this string in place of the current value: **georgia,"times new roman",times.**

Hit OK, and something magical happens. In the original page, the heading displays in our serif font, as you can see in Figure 7.24.

Figure 7.24. Making SitePoint headings pretty with Georgia.



Now, what about centering the text in that box? Well, you'll find that no **text-align** property appears in the available style sheets, so let's go ahead and add one. To do so, right-click anywhere on the Properties pane and select New Property from the context menu. Fill in **text-align** when prompted for a property name, and **center** when prompted for the property's value. When that's done, you'll immediately see the result of your tweaking: the nice, centered, serif font heading shown in Figure 7.25.

Figure 7.25. Centering SitePoint's headings.

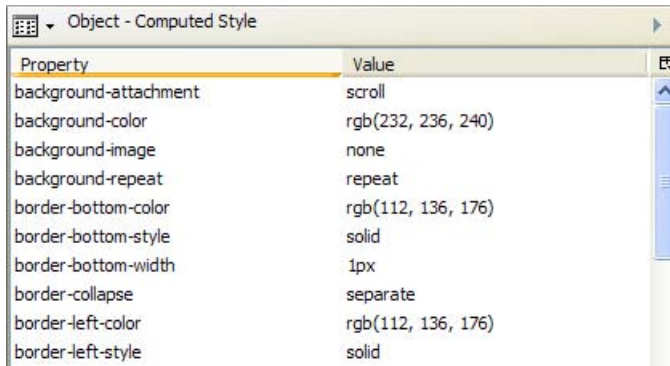


If you like the results, just add the changes to your user-defined CSS file to make them permanent. The DOM Inspector doesn't allow you to change the originating site's content, of course.

The Computed Style View

When multiple CSS rules apply to a single element, Firefox combines them to derive a single set of "computed" CSS style rules. The combination (or "resolution") process is based on the rules of precedence specified in the CSS specification.¹⁴ The Computed CSS Style Rules page shown in Figure 7.26 presents this resolved set of rules.

¹⁴ <http://www.w3.org/Style/CSS/>

Figure 7.26. Reviewing computed CSS style rules.

Using this, and the CSS Style Rules view, you can observe how Firefox resolves CSS rules. Just alter any sheet—or any sheet property—and observe the changes made to the computed result. This is a very practical and experimental way to understand the rules of precedence in CSS.

The JavaScript Object View

The JavaScript Object view lists the JavaScript objects that can be used to script the selected node. That includes all the traditional page features (access to form elements and links), all the DOM features (for example, adding and removing child nodes, and the nodes themselves), and any script content that’s been added by the page designer (such as handlers and extra form object properties). Figure 7.27 provides a glimpse of the numerous object features that are displayed.

A commonly examined JavaScript object is the `target` object, which contains many properties and methods of the selected node, and is particularly useful for finding out which methods are available on the DOM node.

Editing the DOM Dynamically

With the DOM Inspector in place, you can edit the DOM of the document under inspection “live.” This means that you can add or remove HTML content, including scripts and inline styles. Any changes you make are reflected in the document immediately, without the need to reload any files. Figure 7.28 shows the drop-down menu that provides access to these actions.

Figure 7.27. Reviewing available JavaScript objects through the JavaScript Object view.

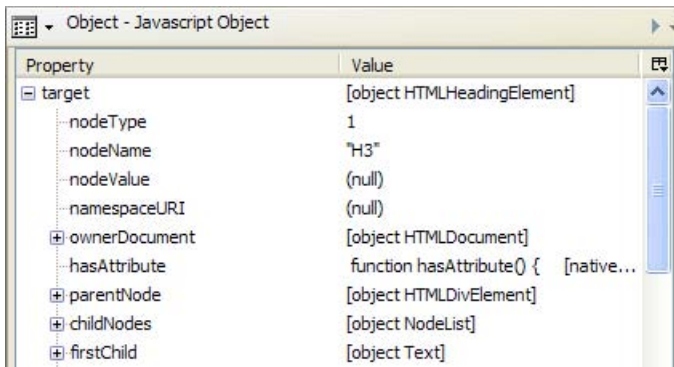
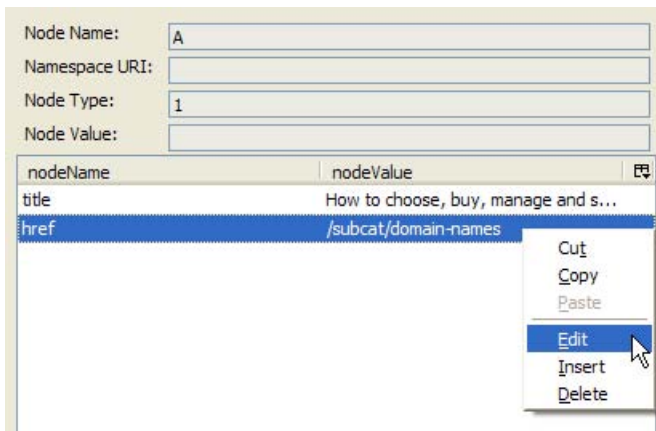


Figure 7.28. Editing the DOM dynamically.



In the DOM view of a document, the child nodes of the selected element are typically its attributes. For example, if an `` element is selected, it would have child nodes `src` and `alt`, among others. So this view is more detailed than a simple hierarchy of tags.

Since you're able to edit and delete existing nodes, and insert new ones, you can use the DOM Inspector to experiment with different attributes for elements within your Web page. And, as for other parts of the DOM Inspector, you can observe the effect "live." For example, you might want to replace the `src` attribute of an `img` tag with another image, thus swapping the images; alternatively, you

might want to toy with table column widths in order to see the effect these changes might have on the columns' proportions.

The DOM Inspector also allows you to inspect XUL pages: the pages that comprise the Firefox browser itself. You can inspect so-called chrome URLs to find out how Firefox's own XUL, JavaScript, and CSS files are coded. Some interesting chrome URLs that are worthy of inspection are shown in Table 7.1.

Table 7.1. Commonly inspected window URLs.

To view this Firefox window...	...try this URL
The browser itself	chrome://browser/content/browser.xul
The Options dialog box	chrome://browser/content/pref/pref.xul
The Bookmarks Manager	chrome://browser/content/bookmarks/bookmarksManager.xul
The JavaScript Console	chrome://global/content/console.xul

Viewing these pages will blow your mind! Such investigation is an important first step for those making the leap into developing XUL applications, extensions, or themes.



Editing the CSS or DOM of a document can sometimes have unexpected results. If, in the midst of your DOM or CSS editing, you find that pages aren't displaying properly or as expected, try restarting Firefox. You may have edited a global property that affects all Web pages, or edited a default Firefox CSS rule. These changes, when made with the DOM Inspector, are temporary; a quick restart will undo your changes and restore the defaults. Editing XUL pages is even more risky: make sure you do your homework before you experiment there.

What's Next?

If you've enjoyed this sample material from *Firefox Secrets*, why not order yourself a copy?

Firefox Secrets will show you, step by step, how to optimize the many hidden and not-so-hidden features of Firefox. You'll learn how to make your browsing faster, slicker, safer, more fun, and—thanks to the limitless customization that Firefox affords—how to make the browser work exactly the way that you want it to.

In the remaining chapters, you'll learn how to:

- ❑ Save time by having Firefox open all of your favorite Websites in tabs at start-up
- ❑ Master the secrets of the *about:config* screen and gain access to hidden browser preferences
- ❑ Make Firefox look exactly as you want it to with CSS
- ❑ Modify any Web page, live in your browser, using the DOM Inspector
- ❑ Troubleshoot common problems with Firefox
- ❑ Maintain your privacy on the Web. Manage the 'cookies' Web browsers store on your computer, and choose which sites are allowed to access those cookies
- ❑ Perform Web searches from the location bar with smart keywords, and add your own customized searches
- ❑ And much more!

You'll also get a free bonus CD-ROM, including the Firefox Web browser itself, the Thunderbird email client, and dozens of hand-picked extensions and themes to improve your browsing experience.

[Order now and get it delivered to your doorstep!](#)

Index

A

- about:config interface, 173–182
 - (*see also* hidden preferences)
 - adding new preferences, 181
 - blinking text removal, 190
 - changing default search engine, 51
 - enabling HTTP pipelining, 180
 - enabling Single Window mode, 177, 179
 - modifying right-click searching, 197
 - Open in Tabs preferences, 81
 - searching from the location bar, 196
- access keys, 215, 245
- accessibility, 250
- ActiveX controls and security, 11
- Adblock extension, 134–137
 - Extension Manager view, 123
 - Macromedia content and, 137
 - testing, 138
- Add Bookmark dialog window, 74, 86
- add-ons (*see* extensions)
- Adobe Acrobat plugin, 66
- advertising (*see* popup windows)
- aggregator sites, Weblogs, 266
- aggregators, RSS, 83
- all.js file, 205–206
- Allowed Sites option, 120–121
- Amazon.com, 39
- anchors, displaying, 245
- animated GIFs, disabling, 241
- annoyance elimination, 10
 - animated GIFs, 241
 - extensions for, 134–141
 - JavaScript disabling, 58
 - noninvasive browsing, 54–60
 - popup blocking, 55–58
 - psychedelic content, 190
 - userContent.css and, 187

- apostrophe, Find As You Type feature, 53

- Apple Quicktime plugin, 66

- Atom news feeds, 84

- Auto Copy extension, 152

- auto-completion

- forms, 7, 94

- search bar, 94

- URLs, using about:config, 181

- URLs, using browsing history, 88, 90

- automatic logins, 96

- “Aviary” branch build, 279

B

- back button

- resizing, 191

- tabbed browsing and, 38

- backups

- profiles, 203

- when using nightly builds, 275

- Begin finding when you begin typing

- preference, 53

- “Best-of-Breed” products, 2, 261

- BetterSearch extension, 146

- blinking text, 190

- blogs, relevant to Firefox, 265, 282

- Boing Boing blog site, 265

- Bonsai reports, 278

- bookmarking

- bookmarks toolbar, 76

- currently open tabs, 75

- opening bookmark folders in tabs, 79

- separators, 78

- using the miniT extension, 119

- bookmarks, 72–81

- automatic backups, 132

- creating, 74

- importing and exporting, 72
 - multiple homepages and, 38
 - not-so-smart Keyword Bookmarks, 75
 - roaming bookmarks extension, 130
 - RSS and Live Bookmarks, 82–88
 - search functionality and, 6
 - Smart Bookmarks, 47
 - Smart Keywords and, 45
 - social bookmarking system, 130
 - sorting within folders, 78
 - synchronizing with IE, 132
 - SyncMarks enhancements, 133
 - Bookmarks Manager, 72, 77
 - manually adding Live Bookmarks, 87
 - Bookmarks menu and Smart Bookmarks, 47
 - Bookmarks sidebar, 79, 190
 - Bookmarks Synchronizer extension, 130, 134
 - Boolean preferences, 174, 181
 - Box Model view, DOM Inspector, 226
 - branch builds, 279
 - broadcast icon, 85
 - browsers
 - (*see also* Internet Explorer)
 - advantages of Firefox, 1
 - alternatives to Internet Explorer, 14
 - free screen space, 9
 - history of Firefox, x, 267
 - importing and exporting bookmarks, 72
 - importing settings from, 21
 - managing multiple Web pages, 27, 29
 - Opera 7.60 user interface, 9
 - popularity and security vulnerabilities, 10
 - raw news feed display, 167
 - tabbed browsing support, 26
 - browser-specific sites, 142
 - browsing history (*see* history sidebars)
 - BugMeNot extension, 140
 - bugs
 - ColorZilla extension, 253
 - Firefox bug reports, 266
 - Open in Tabs feature, 81
 - recommended reporting procedure, 272, 280
 - Bugzilla, 280
 - Burning Edge site, 275
- ## C
- cache
 - clearing, 203
 - disabling, 241
 - calculations within Firefox, 199, 221
 - CD-ROM accompanying the book, xiii
 - certificate-based security, 101, 118
 - character encoding, 213
 - Checky extension, 249
 - chrome subfolder, 186
 - Greasemonkey user scripts, 257
 - profile corruption and, 269
 - chrome URLs, 234
 - ChromEdit extension, 188
 - chunking, 78
 - class attributes, displaying, 245
 - color balance checking, 107
 - Color Palette Browser, 253
 - ColorZilla extension, 251–255
 - page zooming function, 254
 - Compact Menu extension, 198
 - compiling Firefox from source, 20
 - Computed Style view, DOM Inspector, 231
 - configuration (*see* about:config interface)
 - content area (*see* screen area)
 - content filtering, Adblock, 134, 136
 - Content-Type headers, 237

- context menus
 - (*see also* right-click text searching)
 - freeing, 60
 - Greasemonkey extension, 258
 - ieview extension, 141
 - LookAhead Search extension, 149
 - “context-switching pollution”, 29
- context Web searching (*see* right-click text searching)
- Cookie Monster script, 258
- cookies, 90–94
 - disabling, 241
 - displaying set cookies, 247
 - setting preferences, 91
 - Stored Cookies Manager, 93
 - tracking, 218
 - whether delicious or not, 90
- Create Profile Wizard, 201
- createProfile command, 269
- cross-session resuming, 65
- CSS (Cascading Style Sheets)
 - copying a Web color to, 251
 - disabling, 242
 - DOM Inspector and, 14
 - editing on-the-fly, 239
 - Firefox default styles, 228, 243
 - Firefox support for CSS2 and CSS3, 13
 - precedence, 231
 - userChrome.css file, 187
 - userContent.css file, 187
- CSS menu, Web Developer toolbar, 242
- CSS Style Rules view, DOM Inspector, 228
- CSS Zen Garden site, 282
- currency conversion, 48
- customizing
 - appearance with themes, 105–114
 - appearance with userChrome.css, 187

- functionality with extensions, 115
 - setup options, 16
 - toolbars, 68, 191, 193, 197
 - validators, 249

- CuteMenus extension, 168

D

- Data Migration feature, 23, 72

- debugging

- HTML forms, 235, 244

- JavaScript Console and, 13, 219

- Venkman Debugger, 260

- del.icio.us site, 130

- Delicious Delicacies extension, 91

- developers (*see* Web developers)

- development team for Firefox, xi, 265

- DHTML (Dynamic HTML), 257

- dictionary, SpellBound extension, 154

- Dictionary.com

- predefined Smart Keyword, 5, 45

- search string, 196

- Disable menu, Web Developer toolbar, 240

- Document Object Model (DOM)

- DOM Inspector and, 221

- dynamic editing, 232

- document reports, 212

- DOM Inspector, 14, 16, 221–234

- compared to Web Developer toolbar, 239

- editing the DOM dynamically, 232

- as an extension, 222

- information views, 226–232

- selecting nodes and elements, 223

- DOM Node view, DOM Inspector, 226

- dom.disable_* preferences, 176

- Download Manager, 62–66

- disabling, 65

- replacing, 160

- Download Manager tweak extension, 157

- Download Sort extension, 162
- Download Statusbar extension, 160
- downloading files, 61–66
 - configuring preferences, 65
 - defining locations, 162
 - download statistics, 160
 - extensions for, 157–165
 - installing extensions, 121
 - resuming after pausing, 65
- downloading Firefox, 16–23, 271
- downloading nightly builds, 277, 279
- drag and drop operations
 - bookmarking, 74, 78
 - searching, 42, 151
 - tab reordering, 116
 - toolbar customization, 194
- Dragonlance, 39
- drop-down menus
 - DOM editing, 232
 - search engine plugins, 42

E

- Edit CSS extension, 240
- Edit CSS option, Web Developer toolbar, 239
- email
 - Firefox integration interface, 67
 - making addresses clickable, 145
 - Mozilla Thunderbird, xi, xiii
- encryption, 217
- exceptions list, cookie preferences, 92, 94
- Exposé, 29–30, 270
- extension examples
 - Adblock, 134–137
 - Auto Copy, 152
 - BetterSearch, 146
 - BugMeNot, 140
 - Checky, 249
 - ChromEdit extension, 188
 - ColorZilla, 251
 - Compact Menu, 198
 - CuteMenus, 168
 - Delicious Delicacies, 91
 - DOM Inspector, 222
 - Download Manager tweak, 157
 - Download Sort, 162
 - Download Statusbar, 160
 - Edit CSS, 240
 - FeedView, 167
 - FireFTP, 255
 - FlashBlock, 138
 - FoxyTunes extension, 170
 - Google PageRank Status, 148
 - Googlebar, 148
 - GooglePreview, 146
 - Greasemonkey, 257
 - ieview, 141
 - Linkification, 144
 - LiveHTTPHeaders, 235–237
 - LookAhead Search, 149
 - miniT (drag+indicator), 116
 - Mouse Gestures, 155
 - Sage, 165
 - SessionSaver, 127
 - SpellBound, 154
 - Super DragAndGo, 9, 151
 - SyncMarks, 130
 - TargetAlert, 143
 - undoclosetab, 129
 - Unread Tabs, 189
 - Web Developer, 237
- Extension Manager, 123
 - disabling and uninstalling extensions, 124
 - installing from local files, 122
 - miniT installation, 118
 - Safe Mode display, 127
 - Super DragAndGo tuning, 151
 - updating extensions manually, 124
- Extension Room site, 107, 129
- extensions, 9, 115
 - annoyance elimination, 134–141

-
- compulsory registration avoidance, 140
 - disabling, 124, 267
 - distinguished from user scripts, 258
 - file downloads, 157–165
 - Firefox default feature set and, 8
 - for recovery, 127–130
 - fun extensions, 168
 - installing from local files, 121
 - installing from Mozilla Update, 116
 - installing from other sites, 120
 - Internet Explorer fallback, 141
 - mouse operation enhancements, 151–157
 - on CD-ROM accompanying this book, xiii
 - page content information, 142
 - problem fixing in Safe Mode, 125
 - recommended, 127–168
 - roaming bookmarks, 130
 - RSS feed readers, 165–168
 - search functionality enhancement, 146–152
 - troubleshooting problems with, 267
 - updating, 123
 - for Web developers, 9, 14, 234–261
 - Extensions Mirror site, 107
 - extensions, filename (*see* filename extensions)
 - eyedropper icon, 251
- F**
- FastFind feature, 7, 52–53
 - favicons, 28
 - favorites
 - (*see also* bookmarks)
 - bookmarking, 72–81
 - importing from IE, 72
 - loading, using tabbed browsing, 3
 - setting multiple homepages, 37
 - feature set optimization, 8
 - feature strings, JavaScript, 179
 - FeedReader RSS aggregator, 83
 - FeedView extension, 167
 - file download dialog, 61, 121
 - (*see also* downloading files)
 - file formats
 - showing, for links, 143
 - SyncMarks extension, 131
 - filename extensions
 - associating with download file types, 62
 - Download Sort extension and, 162
 - .xpi files, 121
 - filter field, about:config, 175
 - filtering content with Adblock, 134, 136
 - Find As You Type feature, 7, 51–52
 - Find toolbar, 52
 - Firebird (old name for Firefox), 2
 - Firefox 1.0 features (*see specific functions*)
 - Firefox 1.1, 2.0 features (*see* versions, Firefox)
 - Firefox Central, 42
 - Firefox Help Website, 107
 - FireFTP extension, 255
 - FlashBlock extension, 138
 - focus, stealing
 - Download Manager tweak, 159
 - Firefox option, 36
 - with multiple windows, 30
 - folders, user.js location, 206
 - font changes with DOM Inspector, 229
 - Form Manager, 7
 - forms
 - auto-completion, 7, 94
 - debugging, 235, 244
 - LiveHTTPHeader and, 235, 237
 - spell checking entries, 154
 - tab order, 246
 - Forms menu, Web Developer toolbar, 243

- Forms tab, View Page Info, 214
- forums
 - MozillaZine, 9, 264
 - MozillaZine Firefox Builds, 277
 - relevant to Firefox, 266
- forward-slash, Find As You Type, 52
- FoxyTunes extension, 170
- frames
 - enabling resizing, 175
 - inline, Adblock and, 136
- “free scroll” mode, 32
- FTP (File Transfer Protocol)
 - FireFTP extension, 255
 - installable Firefox version, 271
 - SyncMarks extension, 132
- fun extensions, 168
- functionality
 - (*see also* search functionality)
 - customizing with extensions, 115

G

- Gecko layout engine
 - Firefox standards support and, 12
 - render modes, 214
- General tab, View Page Info, 213
- global files, 205
- Google Alerts service, 265
- Google Calculator, 199
- Google keys, 150
- Google PageRank Status extension, 148
- Google search engine, 148
 - Firefox Search bar and, 41
 - “I’m Feeling Lucky” searches, 195
 - predefined as a Smart Keyword, 45
 - right-click text searching on, 51
 - www.google.com cookies, 218
- Googlebar extension, 148
- GooglePreview extension, 146
- Gray Modern theme, 112
- Greasemonkey extension, 257–259

H

- Headers tab, Page Info window, 237
- hidden form fields, 214
- hidden preferences, 173–182
 - (*see also* about:config interface)
 - affecting windows, 175–180
 - forcing frame resizability, 175
 - inline URL auto-completion, 181
 - JavaScript Console and, 220
 - modifying, 175
 - modifying scripted popups, 176
 - setting using about:config, 173
 - Single Window mode, 177
 - speeding up Firefox, 180
 - types, 174
- highlighting
 - errors, JavaScript Console, 219
 - nodes, by DOM Inspector, 225
 - right-click text searching, 50
 - search results, 53
 - source viewer, 210
- history of Web browsers, x, 267
- history sidebars, 88
 - search functionality and, 6
- homepages, 37
 - multiple, using tabbed browsing, 3
- HTML
 - DOM Inspector and, 221
 - dynamic editing, 232
 - elements, finding with DOM Inspector, 225
 - inline display for forms, 244
 - styling selected elements, 229
 - viewing effects of attribute changes, 233
 - viewing source, 13, 210
 - viewing source for elements, 211
- HTML-like export format, 73
- HTTP headers
 - LiveHTTPHeaders extension, 235
 - styling, 236

HTTP pipelining, 180
HTTP POST requests, 48
HTTPS protocol, 100
 View Page Info, 217
hyperlinks (*see* links)

I

“I’m Feeling Lucky” searches, 195
icons
 blocked popups, 56
 ColorZilla eyedropper, 251
 CuteMenus extension, 168
 DOM Inspector, 224
 HTTPS sites, 101
 Live Bookmark-enabled RSS feeds, 85
 plugin content, 66
 tabbed browsing, 28
 TargetAlert extension, 143
 toolbar back, forward and home buttons, 38
 toolbar customization, 194
 toolbar, resizing, 191, 197
 Whitehart theme, 107
id attributes, displaying, 245
IE (*see* Internet Explorer)
IETF (Internet Engineering Task Force)
 standards, 12, 267
iview extension, 141
iFrames, Adblock and, 136
images
 Adblock extension and, 135
 disabling JavaScript for, 60
 disabling with the Web Developer toolbar, 241
 text searching on, 51
Import Settings Wizard, 21–23, 72
Information menu, Web Developer toolbar, 244
information views, DOM Inspector, 226–232

Inspect node icon, DOM Inspector, 224
installing Firefox, 16–23
 DOM Inspector and, 222
 latest-trunk build, 279
 saved profile information, 204
 using FTP, 271
integer preferences, 174
Internet Explorer
 ActiveX controls and, 11
 alternatives to, 14
 iview extension, 141
 multiple windows example, 28
 prospects for IE 7.0, 15, 27
 security vulnerabilities, 10
 synchronizing bookmarks with, 132
 tabbed browsing interface for, 26
 viewing source in IE 6.0, 210
irritating page content (*see* annoyance elimination)

J

Java
 disabling, 242
 plugin for, 66
JavaScript
 Adblock extension and, 135
 disabling, 58, 242
 Greasemonkey user scripts, 257
 modifying scripted popups, 176
 Venkman Debugger, 260
JavaScript Console, 13, 219–221
 calculations, 200, 221
 error display, 219
 Venkman compared to, 260
JavaScript Object view, DOM Inspector, 232
jigsaw piece icon, 66

K

keyboard shortcuts
 creating bookmarks, 74

- FastFind feature, 52
- search functionality, 42
- tabbed browsing, 33
- KISS (Keep It Simple, Stupid) principle, 9

L

- lazy search, 151
- Le Breeze theme, 107
- Linkification extension, 144
- Linkify user script, 257
- links
 - converting plain text to, 144, 257
 - file type indication for, 143
 - opening in “background” tabs, 30–31
 - text searching on, 51, 53
 - visited, 90
- Links tab, View Page Info, 214
- Linux
 - downloading Firefox for, 20
 - downloading nightly builds, 277, 279
 - killing Firefox processes, 269
 - Plastikfox Crystal theme and, 112
 - profile location, 185
 - Profile Manager access, 201
 - reinstalling Firefox, 272
 - Safe Mode operation and recovery, 126
 - uninstalling Firefox, 271
- Live Bookmarks, 82–88
 - creating, 85
 - extension alternative to, 165
 - Sage extension and, 166
- LiveHTTPHeaders extension, 9, 235–237
- location bar searching, 193, 195–196
- logins
 - automatic, 96
 - avoiding compulsory registration, 140

LookAhead Search extension, 149

M

- Mac OS X
 - “context-switching pollution”, 30
 - downloading Firefox for, 18–19
 - downloading nightly builds, 277, 279
 - Exposé and, 29, 270
 - extra windows problem, 270
 - killing Firefox processes, 269
 - opening tabs, 32
 - Pinstripe theme, 105
 - profile location, 185
 - Profile Manager access, 201
 - reinstalling Firefox, 272
 - Safe Mode operation and recovery, 126
 - Saferfox Xpanded theme and, 112
 - uninstalling Firefox, 271
- Macromedia Flash
 - extensions blocking, 137
 - plugin for, 66
- master passwords, 98
- media players, FoxyTunes extension, 170
- Media tab, View Page Info, 216
- media types, styling for, 243
- memory functions, 7, 88–100
 - forms auto-completion, 94
 - history sidebars, 88
 - managing cookies, 90–94
 - Password Manager, 95–99
 - privacy controls, 99
- menubar, maximizing screen area, 197–198
- Microsoft Corporation
 - (*see also* Internet Explorer; Windows)
 - complacency alleged, 14

- middle-clicking, 6
 - back, forward and home buttons, 38
 - bookmark folder Open In Tabs option, 81
 - closing tabs, 33
 - opening a new tab, 32
 - RSS feeds, 165
 - undoclosetab extension, 130
- milestone releases, 274
- miniT (drag+indicator) extension, 116
- Mouse Gestures extension, 155
- mouse operations
 - (*see also* middle-clicking; right-click text searching)
 - extensions to, 151–157
- MozBackup program, 203
- Mozilla ActiveX Project, 12
- Mozilla Application Suite
 - Firefox distinguished from, xi
 - homepage display, x
 - profile for, 183
 - tabbed browsing, 26
- Mozilla Europe site, 264
- Mozilla Firefox Development Charter, 8
- Mozilla Foundation, 264, 267
- Mozilla Navigator as Firefox predecessor, xi
- Mozilla Thunderbird, xi, xiii
- Mozilla Update Website
 - Newest, Popular and Top Rated extensions, 171
 - sourcing extensions, 116
 - sourcing themes, 106
 - Top Rated and Popular Themes, 110
- mozilla.org site nightly builds, 278
- MozillaNews site, 264
- MozillaZine forums, 264, 266, 281
 - extensions from, 9
 - Firefox Builds forum, 277

- multiple windows
 - example using IE, 28
 - stealing focus, 30
 - tabbed browsing alternative, 25, 27
- Mycroft project, 44

N

- navigation toolbar removal, 197
- news
 - (*see also* RSS feeds)
 - about Firefox, 264–265
 - about nightly builds, 275
- newsgroups relevant to Firefox, 266
- nightly builds, 273–279
 - benefits, 274
 - downloading, 277, 279
- noninvasive browsing, 54–60
- non-Web content
 - downloading files, 61–66
 - email integration, 67–68
 - installing plugins, 66–67
- not-so-smart Keyword Bookmarks, 75

O

- Open Folder and Open in the sidebar buttons, 157
- Open in Tabs feature, 80–81
- Open Source status of Firefox, ix, 2
- Opera 7.60 user interface, 9, 26
- opinions about Firefox, 265
- Optimoz extension, 155
- Options panel (*see* preferences)
- Outline menu, Web Developer toolbar, 247

P

- padlock icon, 101
- page display language, 174
- Page Info Security window, 101
- Page Info window, Headers tab, 237
- page zoom, ColorZilla, 254

- PageRank, Google, 148
 - pages (*see* Web pages)
 - Password Manager, 7, 95–99
 - passwords
 - retrieving, 97
 - setting a master password, 98
 - Web Developer toolbar display, 244
 - PDF file detection, 143
 - personalization (*see* customizing; profiles)
 - phishing, 60
 - Phoenix theme, 110
 - Phoenix (Firefox predecessor), 4, 90
 - Pinstripe theme, 105
 - pipe character (|) URL separator, 37
 - Plastikfox Crystal theme, 112
 - plugins
 - adding search engines, 42, 44
 - installing, 66
 - popup blocking, 10, 55–58
 - (*see also* annoyance elimination)
 - customizable, 4
 - disabling, 242
 - user-initiated popups, 58
 - popup windows
 - Firefox 1.1 options, 35
 - modifying using about:config, 176
 - viewing and allowing, 56
 - PopupTest.com site, 179
 - PowerPC/AIX Firefox binaries, 21
 - preferences
 - (*see also* hidden preferences)
 - Adblock extension, 136
 - cookies, 91
 - cross-profile preferences, 205–207
 - file downloads, 62
 - file location, 185
 - Find As You Type feature, 53
 - JavaScript disabling, 58
 - multiple homepages, 37
 - popup blocker, 57
 - precedence, 186
 - saved form information, 95
 - SessionSaver extension, 128
 - software updates, 124
 - storage, 182
 - tabbed browsing, 34
 - Web Features preferences pane, 58
 - prefs.js file, 186
 - preloading Web pages, 3
 - Privacy options panel, 8, 99
 - browsing history retention, 89
 - cookie tracking, 218
 - “profile already in use” message, 268
 - Profile Manager, 200–204
 - Create Profile Wizard, 201
 - deleting a profile, 202
 - restarting Firefox, 204
 - profiles, 182–187
 - backing up, 203
 - component files, 185–187, 270
 - corruption of, 269
 - cross-profile preferences, 205–207
 - editing configuration files, 188
 - folder location, 183–185
 - restoring a saved profile, 204
 - using nightly builds, 275
 - profiles.ini file, 204
 - promoting Firefox, 281
- ## Q
- Quality Feedback Agent, 16
 - QuickFind field searching, 6, 79
 - Quicksearch bookmarks folder
 - creating a Smart Bookmark, 47
 - Smart Keywords and, 45
 - Quirks mode rendering, 214
 - Qute theme, 110
- ## R
- raising and lowering windows, 60
 - Realplayer plugin, 66

-
- reference material relevant to Firefox, 266
 - referrer logging, disabling, 242
 - registration, compulsory, 140
 - regular expressions, 136, 164
 - release builds, 274
 - Render Mode field, 214
 - Requests For Enhancements (RFEs), 280
 - Reset Page option, Web Developer toolbar, 248
 - Resize menu, Web Developer toolbar, 247
 - right-click text searching, 50
 - modifying, 197
 - Web page highlighting, 40
 - roaming bookmarks, 130
 - RSS feeds
 - blogs, relevant to Firefox, 265
 - browser display, 167
 - discovery, 86, 165
 - example feed, 83, 165
 - FeedView extension and, 167
 - Firefox integration, 85
 - Live Bookmarks and, 82
 - MozillaZine and Spread Firefox, 264
 - Sage extension and, 165
- S**
- Safe Mode, 125, 267
 - Saferfox Xpanded theme, 112
 - Sage extension, 165
 - screen area
 - bookmarks toolbar and, 77
 - left free by browsers, 9
 - maximizing, 197
 - screen resolution, 247
 - scroll wheel clicking, 32
 - (*see also* middle-clicking)
 - scrolling marquees, 190
 - Seamonkey (*see* Mozilla Application Suite)
 - search bar, 5, 41
 - auto-completion, 94
 - Google Calculator access, 199
 - LookAhead Search extension, 149
 - removal, 193–197
 - resizing, 192
 - search engines
 - (*see also* Google search engine)
 - adding, 42, 44
 - changing the default, 51
 - searching from the location bar, 196
 - suggestions for Smart Keywords, 49
 - Wikipedia, 44–45
 - Yahoo!, 47, 196
 - search functionality, 39–54
 - calculations, 200
 - different approaches to, 40, 50
 - example sites, 39
 - extensions enhancing, 146–152
 - filtering hidden preferences, 175
 - Find As You Type feature, 7, 52
 - history sidebar, 89
 - keyboard shortcuts, 42
 - modifying right-click searching, 197
 - QuickFind searching, 6
 - right-click text searching, 50
 - search bar, 5, 41
 - searching bookmark titles, 79
 - searching within Web pages, 52
 - Smart Keywords, 45
 - unobtrusive search, 5
 - without the search bar, 195
 - search results
 - highlighting, 53
 - loading into a new tab, 42
 - opening automatically, 149
 - previewing, 146
 - Secure FTP, 256
 - Secure HTTP (HTTPS), 100, 217

- security, 100–102
 - checking for the padlock icon, 101
 - cookies and, 91
 - digital certificates, 101, 118
 - Firefox memory functions and, 88
 - Firefox privacy controls, 8
 - fixes, nightly builds, 274
 - installing extensions, 117, 120, 122
 - locked profiles, 269
 - Password Manager and, 95–97
 - vulnerabilities, 10–11
 - Security tab, View Page Info, 217
 - sensitive information (*see* security)
 - separators, bookmarks, 78
 - SessionSaver extension, 127
 - setup procedure, 16–23
 - custom setup options, 17
 - Shift key, tabbed browsing, 36
 - Show Passwords button, 97
 - sidebars
 - Bookmarks sidebar, 79, 190
 - Download Manager display, 159
 - history sidebar, 6, 88
 - shifting to the right-hand side, 189
 - Single Document Interface (SDI)
 - browsers, 37
 - Single Window mode, 177
 - skins (*see* themes)
 - slang, 46
 - Smart Bookmarks, 47
 - Smart Keywords, 5, 45
 - adding from a search field, 46
 - creating using Smart Bookmarks, 47
 - currency conversion example, 48
 - list of suggested, 49
 - not-so-smart Keyword Bookmarks, 75
 - searching from the location bar, 196
 - social bookmarking, 130
 - Software Installation window, 117
 - Software Update option, 124
 - sorting bookmarks, 78
 - source code
 - (*see also* viewing source)
 - Firefox nightly builds, 273, 277
 - Firefox release builds, 274
 - Sparc/Solaris Firefox binaries, 21
 - SpellBound extension, 154
 - spoofing
 - Internet Explorer vulnerability, 15
 - status bar text, 58, 60
 - Spread Firefox site, 264, 281
 - Standards compliance mode, 214
 - standards support
 - Gecko-based browsers, 12
 - Internet Explorer, 15
 - startup problems, troubleshooting, 268–270
 - status bar
 - Adblock indicator, 134
 - Auto Copy extension, 152
 - Download Statusbar extension, 160
 - preferences, 60
 - stock symbol Smart Keyword, 45
 - Stored Cookies Manager, 93
 - string preferences, 174
 - stylesheets, user-specified, 188–193
 - styling the user interface, 188–193
 - Super DragAndGo extension, 9, 145, 151
 - SyncMarks extension, 130
- ## T
- tab order, forms, 246
 - tabbed browsing, 3, 25–39
 - back, forward and home buttons, 38
 - bookmarking all tabs, 75
 - closing tabs, 33, 36
 - enabling Single Window mode, 177
 - focus option, 36
 - opening bookmark folders in tabs, 79, 81

-
- opening tabs, 32
 - other browsers providing, 26
 - productivity advantages, 30–32
 - rearranging tabs with miniT, 116
 - separating multiple URLs, 37
 - setting user preferences, 34
 - tabs
 - Download Manager display, 159
 - loading with search results, 149
 - marking unread tabs, 189
 - recovering unintentionally closed, 129
 - recovering using SessionSaver, 128
 - tabbed validation results, 250
 - View Page Info feature, 213–218
 - TargetAlert extension, 143
 - taskbar grouping option, IE, 28
 - technical information on Firefox, 266
 - text
 - (*see also* right-click text searching)
 - automatically copying, 152
 - sizing, with ColorZilla, 254
 - spell checking, 154
 - text-align property, 231
 - textboxes, search bar, 41
 - theme examples, 110–113
 - Gray Modern, 112
 - Phoenity, 110
 - Plastikfox Crystal, 112
 - Qute, 110
 - Saferfox Xpanded, 112
 - Whitehart, 107
 - Theme Manager, 106–109
 - themes, 105–114
 - default themes, 105
 - example themes, 110–113
 - installing Whitehart, 108
 - on CD-ROM accompanying this book, xiii
 - sourcing additional themes, 106
 - troubleshooting problems with, 267
 - updating, 109
 - userChrome.css mimicking of, 187
 - thumbnail images, 146
 - Thunderbird email client, xi, xiii
 - toolbars
 - back, forward and home button icons, 38
 - bookmarks toolbar, 76
 - customizing, 68, 191, 193, 197
 - FastFind feature, 52
 - Googlebar extension, 148
 - Web Developer toolbar, 237
 - Whitehart theme, 107
 - Tools menu, Web Developer toolbar, 248
 - topographic information, 246
 - troubleshooting Firefox, 267–272
 - startup problems, 268
 - trunk builds, 279
 - Type-Ahead Find feature, 7, 51–53
- ## U
- undoclosetab extension, 129
 - uninstalling Firefox, 271
 - unit conversions, 200
 - unobtrusive search, 5
 - Unread Tabs extension, 189
 - “Unsigned” extensions, 118
 - Update button, 109, 124
 - Urban Dictionary, 46
 - URLs
 - auto-completion, 88, 90, 181
 - chrome URLs, 234
 - converting to links, 144, 257
 - entering into DOM Inspector, 223
 - javascript: URL, 221
 - separating multiple URLs, 37
 - usability advantages of Firefox, 2–15
 - user interface, 8, 114–115
 - (*see also* extensions; themes)
 - Download Manager, 157
 - email integration feature, 67

- simplifying, 193–200
- styling, 188–193
- user scripts, Greasemonkey, 257–259
- user.js file, 186
 - cross-profile preferences, 206
- userChrome.css file, 186, 189
 - resizing back buttons, 191
 - search bar resizing, 192
- userContent.css file, 187
 - scrolling marquee removal, 190
- user-initiated popups, 58
- usernames, Password Manager and, 7
- users, supporting multiple, 183

V

- validating Web pages, 248–249
- Venkman Debugger, 260
- versions, Firefox
 - development versions, 272–280
 - Firefox 2.0 features, 65
 - Options panel redesign, Firefox 1.1, 34
 - popup options, Firefox 1.1, 35
- vertical bar (|) URL separator, 37
- View Page Info feature, 212
- viewing source, 13, 210
 - JavaScript Console, 219
 - selective viewing, 210
 - Web Developer toolbar, 248
- virus scan, Download Statusbar, 161

W

- W3C (World Wide Web Consortium)
 - standards, 12, 267
- Web address conversion to links, 144, 257
- Web Developer extension, 9, 237
- Web Developer toolbar
 - cookies and, 241, 247
 - CSS diagnostics, 242
 - disabling styles, 242

- disabling unwanted features, 240
- editing CSS on-the-fly, 239
- Information menu, 244
- validation with Checky, 249
- Web developers
 - extensions relevant to, 9, 234–261
 - Firefox advantages for, 2–15
- Web development
 - cookie tracking, 218
 - DOM Inspector, 221–234
 - Firefox’s standard tools, 210–234
 - JavaScript Console, 219–221
- Web Features options dialog, 10
- Web pages
 - accelerating download, 180
 - Auto Copy from, 152
 - copying colors from, 251
 - editing CSS on-the-fly, 239
 - grouping tabs, 27
 - managing multiple pages, 3, 27, 29
 - manipulation with user scripts, 257
 - multiple homepages, 37
 - “multi-view display”, 223
 - opening in “background” tabs, 30
 - revisiting, 71
 - searching within, 52
 - userContent.css styling, 187
 - validation, 248
 - View Page Info feature, 212
 - viewing source, 13, 210, 248
 - viewing source for elements, 211
 - viewing using DOM Inspector, 223
- Web servers, 11
- Weblogs, relevant to Firefox, 265, 282
- Websites relevant to Firefox, 264
- Whitehart theme, 107–108
- whitelists, 57, 154, 259
- Wikipedia
 - adding search engines, 44
 - Firefox on, 267
 - predefined as a Smart Keyword, 45
- wildcards, Adblock extension, 136

window resize grip, 56
Windows Media Player plugin, 66
Windows Task Manager, 268
windows, Firefox
 (*see also* popup windows)
 alternatives to multiple windows,
 25, 27
 ensuring closing, 108
 hidden preferences affecting, 175–
 180
 inspecting coding of, 234
 Mac OS X extra windows, 270
 resizing, raising and lowering, 59
Windows, Microsoft
 “context-switching pollution” in XP,
 30
 downloading Firefox for, 16
 downloading nightly builds, 277,
 279
 email integration feature for, 67
 Phoenixity theme and XP, 110
 problems restarting Firefox, 268
 profile location under 95, 98, 98SE
 and ME, 184
 profile location under NT, 185
 profile location under XP and 2000,
 183
 Profile Manager access, 201
 Safe Mode operation and recovery,
 125
 uninstalling Firefox, 271
Winstripe theme, 105

X

XBL Bindings view, DOM Inspector,
 227
XML Bookmark Exchange Language
 (XBEL), 132
XUL pages, 174, 234

Y

Yahoo! search engine
 adding a Smart Keyword, 47–48
 search string, 196

Z

“zippy” files, 122